

# Should a company enforce the use of a single architecture across all development efforts?

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

Viewed 199 times

---



0

I know it's a slightly rhetorical question, but I'd like to hear opinions that support or reject this doctrine as it will help me build a better case in my request to not use the dictated solution.



A little background: I develop and maintain a large, mature application (VFP for the UI, Oracle PL/SQL for the middle and back end) that is used strictly in-house. I have asked my superiors about re-writing the UI in C#, but I've been told that all future development efforts are to be done in Java /Spring. I've explained that the amount of effort going from a desktop application to a web application would be significantly larger than simply migrating to C#. I have also explained how the whole UI would have to undergo a major redesign to make the transition to a web browser. Finally, I explained that this application is only used in-house so there wouldn't be as much benefit realized by having a web architecture in place as opposed to the benefit an external application would receive. Unfortunately my arguments have not swayed them.

Part of me is tempted to capitulate and attempt the rewrite in Java just so I can gain the experience, but I fear it could be a disaster of a project that will take quadruple the time and effort that a migration to C# would have required.

I understand that there are benefits to having a department support only a single language, but to me it feels like we're trying to drive screws in with a hammer.

So, what are the positives and negatives of a company using a single solution for all development efforts? Also, does anyone else find themselves in a similar situation where their language of choice is rejected despite all the practical reasons for that choice?

architecture

user-interface

doctrine

Share

edited Feb 12, 2009 at 14:16

Improve this question

Follow

asked Feb 12, 2009 at 13:58



Barry

492 ● 1 ● 4 ● 12

5 Answers

Sorted by:

Highest score (default)





2

To be fair, the question as stated is something of a straw man argument. Of course the answer is **no** if we take the "no matter the cost" condition literally.



**EDIT:** OP has removed the "no matter the cost" wording, the rest of my answer should still apply though.



That said, usually the reason for enforcing a single architecture is to save costs. They want everyone to devote their efforts into that architecture. If that architecture fits most of the problems they deal with, this can reduce their costs. The question then becomes: Is the savings of enforcing an architecture worth the cost of occasionally using it for problems where it isn't appropriate?

There are going to be many instances where the answer is still no, but isn't black-and-white, it depends on what your company does, where your core competency lies and what projects they are really interested in developing.

Share Improve this answer

edited Feb 12, 2009 at 14:18

Follow

answered Feb 12, 2009 at 14:13



[Adam Bellaire](#)

110k ● 19 ● 152 ● 165

---

Good point. I removed the "no matter the cost" to make this a less biased discussion. – [Barry](#) Feb 12, 2009 at 14:17

---

Okay, I've reflected that in my answer. – [Adam Bellaire](#) Feb 12, 2009 at 14:19

---



1



I agree with you and Rob Wells that it is a bad idea to enforce a single architecture doctrine. The only thing that might work in this case is to do an analysis of cost of each solution (Java Vs. C#) and make a presentation to them with actual numbers and conclusive evidence. You can also try to see if there are any case studies. If they are unavailable, you might want to see if any other company has done something similar, study its effects and use that to convince your management. That said, this may be more work than coding in Java (Kidding :)). I wish you luck in convincing your management.

Share Improve this answer

answered Feb 12, 2009 at 14:09

Follow



[batbrat](#)

5,213 ● 3 ● 33 ● 38



0



No.

That would be like going to an architect who can only work with concrete.

You can only superimpose a single solution to all problem spaces.

**Edit:** Mind you, it would be **very** difficult to convince any management that worthwhile ROI will come from the rewrite of a mature system.

Share Improve this answer

answered Feb 12, 2009 at 14:01

Follow



Rob Wells

37k ● 13 ● 84 ● 147

---

This management hates that it's in Foxpro--it's almost like an embarrassment to them--so they are not opposed to a rewrite. – [Barry](#) Feb 12, 2009 at 14:06

---



0



An architecture is not usually imposed to make one project team's life easier, so unless you were using Java/Spring before it's going to come at some cost to you.

A choice of Java and webapps isn't necessarily just about a single language.

There are many reasons why the company may mandate a single architecture. You need to understand the reasons why your company has done that and the value to the company from doing so.

The reasons for doing so might include

- Having to maintain expertise train people in only one language and toolchain
- Making it easier to move people between projects, since the technology is similar.
- Making maintenance easier, since staff working on the application later don't have to know multiple languages.

- Not having to buy tools (IDEs, refactoring browsers etc) for multiple languages
- Making it easier to use non-Windows desktops
- Reducing the operational costs of supporting the application (e.g. no need for desktop installations, or through reducing the number of technologies they need to know)
- IT Director is getting backhanders from Java tools vendors :)

So there are benefits, as well as costs, in adopting the architecture. You need to show that you at least understand the benefits and can make the case why using your preferred language on this project makes corporate sense.

Why do you think it will be a disaster? What make Java such a bad fit for the rewrite? You need to be careful you're not coming over as someone who has a favourite language and just bad mouths things you don't yet know.

Share Improve this answer

Follow

answered Feb 12, 2009 at 14:26



[The Archetypal Paul](#)

41.7k ● 20 ● 106 ● 135

---

I love JAVA with the limited exposure I've had to it so far. It's not the language, it's the cost of converting a mature app with a user-base that's comfortable with the existing version to a significantly different architecture (web) that's making me fear a disaster. – [Barry](#) Feb 12, 2009 at 14:57

---

OK - the impression I got from your posting was the the language was a significant issue. You may have inadvertently given that impression to your enterprise rule enforcers too? Have your architecture people mandated web applications? If so, do you know the reasons for that? – [The Archetypal Paul](#)  
Feb 12, 2009 at 16:34

---

I think you second bullet (Making it easier to move people between projects, since the technology is similar.) hit management's main reason for enforcing one architecture. Allowing me to develop in C# would ultimately require them to hire more developers. – [Barry](#) Feb 12, 2009 at 17:44

---



0

Limiting the choice of "sanctioned" architectures makes sense if you consider maintenance costs. It makes it easier to move people or even entire teams between projects.



Share Improve this answer

answered Feb 12, 2009 at 14:32



Follow



[nyenyec](#)

7,148 ● 9 ● 31 ● 15

