

Improving the quality of code? [duplicate]

Asked 16 years, 2 months ago Modified 16 years, 2 months ago

Viewed 1k times



10



This question already has answers here:

[Guidelines to improve your code](#) (21 answers)

Closed last year.

So, in reading this site, it seems that the shop in which I work does a lot of things wrong and some things right. How can I improve the code that I work with from my colleagues? The only thing I can think of is to lead by example - start using Boost, etc. Any other thoughts?

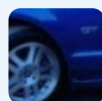
C++

Share

Improve this question

Follow

asked Oct 2, 2008 at 0:32



[jdt141](#)

5,133 ● 7 ● 37 ● 37

You may want to narrow this down some. Thousands of books have been written on this subject. :) – [Jason Baker](#) Oct 2, 2008 at 0:35

Lead doing what by example? Any examples of examples you'd lead with? – [S.Lott](#) Oct 2, 2008 at 0:49

+1 for "lead by example" – [Baiyan Huang](#) May 31, 2013 at 23:29

12 Answers

Sorted by:

Highest score (default)



3



You probably have to look more closely at what it is your shop does wrong and what they do right. What can you actually change there? *What can you change about your own practices that will improve your skills or that of your team?* It can be difficult to realize change in an entrenched shop. Try proposing code reviews (on your code first), which could lead to discussion.

For tangible items, I'd look at [Scott Meyers](#)' Effective C++, etc. Develop your skillset and you will either help improve others around you or move on to a shop that will. Also, look at the Gang of Four's [Design Patterns](#) book.

Share Improve this answer

answered [Oct 2, 2008 at 0:38](#)

Follow

community wiki
[Kris Kumler](#)

I'll also note, this answer was quick and could have gone on forever, considering the breadth of the question. I'm assuming this is someone without too much experience in different shops at all. – [Kris Kumler](#) Oct 2, 2008 at 0:39



2



Code reviews are the best way I found to improve code quality overall. Reviewing code from different individuals helping each other increases general awareness of different techniques and help propagate best practices. Hire a person more experienced than you are is also a good tool but it is a bit more tedious to implement.



Share Improve this answer

answered Oct 2, 2008 at 0:49

Follow



[David Segonds](#)

84.7k ● 10 ● 49 ● 66



2



- Reading good programming books
- Learning from other's code - Open source projects are the best place to start
- Read good blogs and forums regularly - Sutter mill, Coding Horror, Martin fowler etc
- Code reviews
- Unit tests
- Using good libraries like Boost, STL. Also understanding their implementation



Share Improve this answer

answered Oct 2, 2008 at 5:43

Follow



TG.

1,035 ● 3 ● 15 ● 23



Other things to try is to add [unit tests](#) and [documentation](#).

1

Share Improve this answer

answered Oct 2, 2008 at 0:42

Follow



1800 INFORMATION

135k ● 30 ● 163 ● 242



0

Leading by example is always a good thing, though convincing others that your example is better than however they're currently doing it is not so easy.



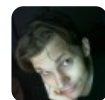
Constructive criticism through code review is probably your best bet for gently suggesting alternative approaches to how your colleagues work. The key point is to convince others that what you're proposing really is better in a tangible way that they can appreciate.



Share Improve this answer

answered Oct 2, 2008 at 0:35

Follow



Jason Etheridge

6,897 ● 5 ● 31 ● 33



0

Sometimes folks have to see that your way is working better than their way. It is often difficult to make people change.



Have you considered unit test writing if you don't do that already? I've found it to really improve my production code and give me more confidence that what I'm writing is what I'm supposed to be writing.



I like Jason's idea about code reviews. They *can* be helpful or they can be a place for arguing - really depends on how you set the tone.

Share Improve this answer

answered Oct 2, 2008 at 0:39

Follow



[itsmatt](#)

31.4k ● 11 ● 102 ● 165



0



Although this probably isn't as direct of an answer, I recommend you pick up the book [Code Complete](#). I find it to be the best resource for learning how to be a better programmer. If you read through the whole book and understand what it talks about, you'll really learn how to better yourself, and your code.



Share Improve this answer

answered Oct 2, 2008 at 0:44

Follow



[Dan Herbert](#)

103k ● 51 ● 192 ● 221



0



Architect and design the project well so that none of the developers will be able to take a different route to violate the quality. If you set a great design, people will just follow the route and they will automatically learn

Share Improve this answer

edited Oct 2, 2008 at 0:46



Follow



answered Oct 2, 2008 at 0:39



[Jobi Joy](#)

50k ● 20 ● 110 ● 120



0

I find writing unit tests helps code quality a lot - it means you have to think about how your code will interact with the tests and other parts of the code.



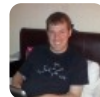
Peer code-review: Checking quality of code will also make the programmers think about how they write the code.



Share Improve this answer

Follow

answered Oct 2, 2008 at 5:26



[graham.reeds](#)

16.5k ● 17 ● 75 ● 138



0

It's great that you recognize that there's room for improvement and have the desire to try to enact some change. I suggest reading James Shore's [19-week diary](#) where he documents the steps that he went through to enact agile development at his company. It's a hard fight, but his experience shows that you can make a difference.



Share Improve this answer

Follow

answered Oct 2, 2008 at 5:40



[Mike](#)

1,594 ● 11 ● 12



Just asking the question is a good start.

0

Specifically you can:



- Admit that your code sucks
- Start asking others, preferably others with more experience, to review your code
- Implement a continuous build server - you have to be the one who uses this first
- Have courage because this can be difficult
- Be humble
- Read Code Complete
- Use a software development methodology that encourages team work. Some of the agile methodologies are really good at this
- Read development blogs
- Get involved in a user group



Change is hard and you have to be the one who changes first.

If you are working in an environment where others are happy the way it is you are going to have rough going. Be persistent about wanting to improve code quality.

Share Improve this answer

answered Oct 2, 2008 at 13:07

Follow



SWD

289 ● 1 ● 5 ● 13



0



I am biased (as a result of my work), but depending on your budget (if it exists) static analysis is a possible option. There are lots of different types of tools, some of which also include coding standard enforcement checking..



If you use g++, you might be able to get a basic amount of help from the `-Weffc++` option.



Share Improve this answer

answered Oct 3, 2008 at 12:53

Follow



[Richard Corden](#)

21.7k ● 9 ● 61 ● 87
