How would you go about evaluating a programmer? [closed]

Asked 16 years, 4 months ago Modified 10 years ago Viewed 4k times

▲

10

▼

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 13 years ago.

A few weeks ago, I was assigned to evaluate all our programmers. I'm very uncomfortable with this since I was the one who taught everyone the shop's programming language (they all got out of college not knowing the language and as luck would have it, I'm very proficient with it.). On the evaluation, I was very biased on their performance (perfect scores).

I'm glad that our programming shop doesn't require an average performance level but I heard horror stories of shops which do require an average level.

My question are as follows:

1. As a programmer, what evaluation questions would you like to see?
2. As a manager, what evaluation questions would you like to see?
3. As the evaluator, how can you prevent bias in your evaluation?
4. I would love to remove the evaluation test. Is there any advantages to having an evaluation test? Any disadvantage?

evaluation

Share

Improve this question

Follow

edited Dec 15, 2014 at 16:07

MysticMagicϡ

28.8k ● 16 ● 75 ● 125

asked Aug 18, 2008 at 2:29

MrValdez

8,613 ● 10 ● 57 ● 79

4 Answers

Sorted by: Highest score (default)

▲

11

▼

Gets things done is really all you need to evaluate a developer. After that you look at the quality that the developer generates. Do they write unit tests and believe in testing and being responsible for the code they generate? Do they take initiative to fix bugs without being assigned them? Are they passionate about coding? Are they always constantly learning, trying to find better ways to accomplish a task or make a process better? These questions are pretty much how I judge developers directly under me. If they are not directly under you and you are not a direct report for them, then you really shouldn't be evaluating them. If you are assigned in evaluating those programmers that aren't under you, then you need to be proactive to answer the above questions about them, which can be hard.

You can't remove the evaluation test. I know it can become tedious sometimes, but I actually enjoy doing it and it's invaluable for the developer you are evaluating. You need to be a manager that cares about how your developers do. You are a direct reflection on them and as they are of you. One question I always leave up to the developer is for them to evaluate me. The evaluation needs to be a two lane road.

I have to also evaluate off a cookie cutter list of questions, which I do, but I always add the above and try to make the evaluation fun and a learning exercise during the time I have the developer one on one, it is all about the developer you are reviewing.

Share Improve this answer Follow

answered Aug 18, 2008 at 3:34

Dale Ragan
18.3k ● 3 ● 55 ● 71

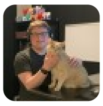
I would first consider not necessarily the number of lines of code, but the value of the code that the person adds as

2

reflective of course to what they are assigned to do. Someone told to maintain code verses building a new app is very different. Also consider how the person uses new techniques to make the code relevant and updated? How maintainable is the code the person creates? Do they do things in a manner that is logical and understandable to the rest of the team? Does their coding improve the app or just wreck it? Last and not least does their coding improve over time?

Share Improve this answer Follow

answered Aug 18, 2008 at 3:06



Justin Yost

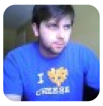
2,340 ● 19 ● 31

0

What about getting everyone's input? Everyone that a person is working with will have a unique insight into that person. One person might think someone is a slacker, while another person sees that they are spending a lot of time planning before they start coding, etc.

Share Improve this answer Follow

answered Aug 18, 2008 at 2:38



Ryan Fox

10.2k ● 5 ● 39 ● 48

0

> What about getting everyone's input? Everyone that a person is working with will have a unique insight into that person.

That would work if (1) evaluation is conducted with open doors and (2) you've worked with that person on one project or even on the same module. As the person evaluating them, I couldn't judge the programmers who I didn't directly work with.

> One person might think someone is a slacker, while another person sees that they are spending a lot of time planning before they start coding

Unfortunately, this is debatable. Someone who looks like a slacker might be in deep thoughts, or maybe not. And is someone who spend a long time planning, necessarily a bad programmer?

I believe a good evaluation question would be able to answer this.

Share Improve this answer Follow

answered Aug 18, 2008 at 3:05

MrValdez

8,613 ● 10 ● 57 ● 79