# What to do with star developers who don't document their work? [closed]

Asked 15 years, 10 months ago Modified 7 years, 5 months ago

Viewed 7k times

▲

43

▼

**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 7 years ago.

Improve this question

There is a colleague who seriously knows his stuff, he is one of the brightest I've ever worked with, but he:

- works in his own little area of his home directory rather than in the common CVS repository
- doesn't **document** his code
- doesn't **comment** his code, e.g. 3,500 SLOC of C with no comments and no blank lines to break things up
- often overcomplicates things, e.g. uses three shell scripts that call one another to do the work that one

simple shell script could do.

Maybe this possibly is one of those people who thinks "if I'm the only person who knows this, they can't get rid of me"?

Any suggestions on what to do?

BTW Management knows about the situation and are trying to change things.

**collaboration**

Share

Improve this question

Follow

edited Feb 7, 2010 at 2:49

community wiki
7 revs, 3 users 90%
Rob Wells

2 Even if he is the brightest, he doesn't sound as a team player and as such is of very limited value for the team (and the company). – Toon Krijthe Feb 10, 2009 at 13:46

10 I think the right word is "diva", not "star" – Jon Limjap Feb 10, 2009 at 16:52

8 This is definitely NOT a "star" developer, IMO. -R – Huntrods Feb 10, 2009 at 20:40

6 I can understand a good dev who might be a bit stubborn in his ways, but when I read the part "often overcomplicates

things" a big red flag went up, not a "star" in my book, perhaps he just works very hard to get a lot done. He oculd likely be more efficient. – Neil N Apr 2, 2009 at 19:22

1 Lets call them rats, reverse star programmers. – Toon Krijthe Oct 28, 2010 at 11:22

## 40 Answers

Sorted by: Highest score (default)

1 2 Next

126

✓

In my opinion somebody doing such stupid things as you have described above can't be a star developer! To me it seems like he intentionally makes things more complicated as they are, so that nobody else than himself can maintain the code. This makes himself more important than he really is! Talk to him. He has to change it! If he doesn't, replace him with a real star-developer!

I promise you, in even half a year he will not know how his own code works! Fire him and you can save a lot of time and money.

Share Improve this answer Follow

edited Feb 10, 2009 at 14:07

community wiki
2 revs
Alexander

1 Classic case of writing into job security. – David Feb 10, 2009 at 19:00

Umm, do any of us truly remember what we did 6 months ago? I know I don't. I agree with the other points made here though. – OutOFTouch Feb 11, 2009 at 15:09

2 Can't agree with that. The code you are writing is not your personal property, it's the property of your company! So you must ensure the property of your company doesn't get lost because of the foolishness of a single person. – Alexander Feb 11, 2009 at 20:03

1 @OutOfTouch, which is exactly why we need to create elegant code so we don't need to take days to go through it and figure out how it worked. – Hans Sjunnesson Feb 13, 2009 at 14:50

Calm down with those exclamation marks! – xian Jun 22, 2009 at 6:57

46

The CVS part is easy - an "accidental" hard drive failure will teach him a lesson for a life (make sure you have a backup though so you won't actually lose code)

Share Improve this answer Follow

answered Feb 10, 2009 at 13:39

community wiki
Tamas Czinege

2 This is negative, not positive. How can it help. This is what you should do as a last resort before you consider the firing option. We don't actually know if this guy is going out of his way to do this, or if he's just never been told! – Robin Day Feb 10, 2009 at 15:26

2 @Robin:Why don't you ask him then? Like "Listen mate, you're really great but the rest of the team struggles reading your code so it would be great if you could..." etc. – Tamas Czinege Feb 10, 2009 at 15:30

2 @DrJokepu: My point exactly, this question states three issues with the developer in question with absolutly no indication of how any of them have been addressed already. If they haven't then its there's no debate. He could easily turn around and say. "Sorry, I'll do from now on." – Robin Day Feb 10, 2009 at 15:37

2 It happend. Nothing changed. Star developer blamed IT staff. It did not teach him anything – LicenseQ Feb 10, 2009 at 16:44

5 He's not a star developer, then. He's just a smart ego maniac. – Richard Levasseur Feb 10, 2009 at 18:05

---

▲

26

▼

That sounds like a tough situation.

Personally, I would let him go. He may be a star developer, but he isn't a team player. And you need to have a cohesive team that can work together if you want to make a good product.

Share Improve this answer Follow

answered Feb 10, 2009 at 13:39

community wiki
DarthNerdus

1 I might argue that he isn't a star developer because he isn't working well with the team. Also, superstars don't

overcomplicate-- pragmatism rules the day. – Greg D Feb 10, 2009 at 14:10

Suggesting letting him go seems a little extreme when all you know is what is written above. I would be wary of any manager that would make such a decision so lightly. It quite rightly takes many stages in order to get rid of someone. Tell them to do X, then if they don't, then you have a case! – Robin Day Feb 10, 2009 at 15:30

20

Failing to document is a (very bad) way to ensure job security.

You can do several things to counter this:

- add documentation as a requirement for the personal performance reviews.
- don't accept software that isn't documented.
- have a word with the developer and find out why he doesn't document.
- Buy a cool documentation tool.

Share Improve this answer Follow

answered Feb 10, 2009 at 13:45

community wiki
Toon Krijthe

Play the **bad cop/good cop** sketch you have seen from the movies. Let the management be the bad cop and you

17 be the good cop. Let the managament ask for over-kill documentation and per-minute ZIP backups of his work. But you offer him moderate documentation (by doxygen for example) and usual source control check-ins...

Share Improve this answer Follow

answered Feb 10, 2009 at 14:09

community wiki
Malkocoglu

---

Talk to him?

11 If he really is a "star developer" he'll take note of what you say.

It may not change him overnight but it might be that he is just completely unaware that other people dont get it quite like he does.

---

Edit:

It's probably a bit late to change now, but more information is needed in working out a solution. It's impossible for anyone here to actually suggest letting the guy go based on these points alone. If you've been telling the guy every day for the last year that he needs to change or he's out of here, then you can let him go. However, I see no evidence of that.

A brilliant developer can be taught to use source control, comment and document. If you spend the effort here then you truly will have a star developer.

Share Improve this answer Follow

edited Feb 10, 2009 at 15:35

community wiki
2 revs
Robin Day

---

▲

10

▼

You might be focusing on the wrong area here, you are being provided with an opportunity to see some weaknesses in your process.

> - *works in his own little area of his home directory rather than in the common CVS repository*

A simple chat will probably suffice here, the benefits of version control speak for themselves and any "bright" person would likely be enthusiastic about those benefits. However it may also be a good chance to look into alternative version control systems that allow greater ease of use and flexibility (take a look at bzr and git). Better yet, get him involved in the selection process, if he really is a "star" he'll probably have good input and be more vested in its use.

> - *doesn't document his code*

It doesn't sound like documentation is part of your process. People are going to resist having to do extra work and if there isn't a defined process then you're talking about a lot of extra work. Is documentation really needed? If so, is there a process defined for creating it? should you have someone entirely dedicated to it? Should you at least have a tool to facilitate it (maybe something as simple as mediawiki)?

> - *doesn't comment his code, e.g. 3,500 SLOC of C with no comments and no blank lines to beak things up*

Three words: peer code review. Outside of obvious error-catching benefits, this also can provide some peer pressure, which is a strong force and can be a good thing. Wanting to be perceived well by your peers self-generates ownership and quality.

> - *often overcomplicates things, e.g. uses three shell scripts that call one another to do the work that one simple shell script could do.*

Again, peer code review. You mention that management knows about this programmer's deficiencies. Does he? It's rather difficult for people to change and improve if they don't recognize the problems with the way they're doing things.

And, perhaps best of all, by coming up with plans to improve your development process (which will likely improve not only your "star" but everyone else on the team) you can probably earn some gold stars for yourself from management.

Share Improve this answer Follow

answered Feb 10, 2009 at 17:49

community wiki
MarquisDeMizzle

9

Doesn't sound like much of a star programmer to me. All the good programmers know that code formatting and use of source control matters. Sounds like although he makes good progress by himself, he's obstructing the progress of the other team members, which might have a net negative effect on the work getting done. Talk to him, and if he refuses to change his practices, let him go.

Share Improve this answer Follow

answered Feb 10, 2009 at 14:12

community wiki
Kibbee

8

If he really is that bright and you cannot change his ways, nor do you want to lose him but you still want your code to be documented and commented then my suggestion would be to let a less experienced developer do the documenting and commenting for him. Personally if I were a star developer I would feel prettiy foolish if someone else was made to comment my code and I would start to do it myself eventually. In the meantime while that does not happen the less experienced developer may learn a thing or two.

Share Improve this answer Follow

answered Feb 10, 2009 at 13:54

community wiki
Hans

---

7

There's more to being a star developer than just being an excellent programmer. If he doesn't have team skills and is purposefully ignoring team standards it needs to be brought up to him. If he refuses to adhere to them after talking with management, perhaps he's not the right fit for your company.

Share Improve this answer Follow

answered Feb 10, 2009 at 13:49

community wiki
Ryan Thames

Agreed. Being a star developer to me implies he can and does do it all, as well as being and example of how things should be done to the other programmers. If he's not doing this as a matter of course, there is nothing "Star" about him. – Aikislave Feb 10, 2009 at 14:45

6

This question makes me nervous, because while the person you're describing sounds egregious, I can see a little bit of myself in him.

I think I'm a pretty good team player, and I'm lucky to be on a very good team. However, I do use methods that my colleagues don't understand, though I've tried pretty hard to explain them. There just is a pretty large experience gap, which is no reflection on any of us.

Documentation is a broad and tricky subject. I try to follow the DRY (don't repeat yourself) maxim. Documentation that is separate from the code can amount to repeating yourself, so it is liable to get out of date unless you slow yourself down to keep it up to date. My usual approach is to touch it up afterward.

Often the problem I'm working on is so tricky that I can advance plan and document all I want, but when it comes to the code I often discover that I was wrong and have to re-think it. So the idea that you can document stuff in advance, and just follow that, only works for pretty straightforward problems, it seems to me.

Anyway, I think this is an excellent question, and the answer is not at all simple.

Share Improve this answer Follow

answered Feb 10, 2009 at 14:48

community wiki
Mike Dunlavey

very well thought out +1 – Steve Feb 11, 2009 at 1:01

I second this notion. Without more context, it would be hard to provide a cut/dry suggestion, despite the OP's obvious pre-determined opinion that the developer is a "problem". Not saying he is or isn't, but there could be valid reasons for each of those behaviors. – Nick Feb 11, 2009 at 3:33

5

Is the guy really a rock star? Seriously? Think about it for a second. Is he smart, but doesn't get things done, or is he both smart and able to get things done?

Think about it really hard.

If he really is a rock star, then maybe you shouldn't mess with him. He's producing incredibly awesome things using his own process. Just because there's a different way of doing things that works best for you, doesn't mean that's going to enable him to produce his best work. Instead of trying to get him to bend to your process, which very well could kill all of his awesomeness, you should try and find a way to accommodate the way he works.

If he really is as good as you say, you shouldn't mind doing that. If it isn't worth the effort to do that, then he really isn't that good. In that case, you don't have a rock star, you just have a mediocre programmer that doesn't like to play be the rules. Those guys, you should just get rid of. A temperamental rock star is usually worth the pain, though, because of the quality of what he or she can produce. Those people, you should go to great lengths to keep.

Share Improve this answer Follow

answered Feb 10, 2009 at 13:56

community wiki
Scott Wisniewski

@Scott, it's not so much trying to bend him to a new process. I's just increasing our truck number just in case, God forbid, he got hi by a truck. ATM we would not be able to do several business critical parts of the business. 8-O – Rob Wells Feb 10, 2009 at 14:08

If he's that important and there is a risk of losing him if you try to change him. Maybe the key is to hire someone to do the "dirty work" for him. Have someone clean up after him on a regular basis. Don't get rid of Smart and Gets things done due to some flaws however big they may seem. – Robin Day Feb 10, 2009 at 15:46

Sounds like a star programmer who's bored of his job and is over complicating things to make it more of a challenge. He'll find something better soon enough.

4

Share Improve this answer Follow

answered Feb 11, 2009 at 3:24

community wiki
vanja.

3

Trying to change things? What do you prefer, a poorly documented working piece of software or a well documented junk? Some people is capable of writing software that requires little to no comments, that is not a reliable indicator of quality.

I'm afraid you're going to lose a good developer.

Share Improve this answer Follow

answered Feb 10, 2009 at 13:40

community wiki
Otávio Décio

@ocdecio, sorry mate, it's not that simple. You've picked two silly extremes there. I want good software that doesn't take ages to get an idea of what it does. Some documentation is always helpful. – Rob Wells Feb 10, 2009 at 13:50

A good developer can do both. So why paint everything in black and white? – Dan C. Feb 10, 2009 at 13:51

@Rob: did you talk to your colleague? If so, what does he have to say? – Otávio Décio Feb 10, 2009 at 13:56

3

"Hi Star Developer,

just a little informal heads-up to tell you that from next week, we'll be requiring documentation of code, and helpful commenting in the code - it's going to be company policy, and there'll be no exceptions"

From then on, you just deal with that failure the same as you would deal with a failure to turn up on time, a failure to stop goofing off during work, etc. Bottom line is if the boss says document, you document or you're not doing your job properly.

Share Improve this answer Follow

answered Feb 10, 2009 at 13:56

community wiki
grunties

1 Who gives a damn if a programmer arrives on time, leaves on time, or goof off during working hours? What really matters is how much efficient code the guy produces. Running a business through orders isn't a good idea, especially not in a field were initiative and smarts are a key requirement. – Sylver Feb 10, 2009 at 17:15

Sounds great if you're trying to ruin his morale and push him to quit. If you just want to get him to do the right thing, the carrot is far more effective than the stick. – Orion Edwards Feb 11, 2009 at 0:58

3

If he is working like this, he is not a star developer - great software developers understand that maintainability is extremely important. You will probably pay dearly for this in the long run, I would be very direct with him about how serious this is and let him go if he can't start to adjust. I have seen this plenty of times before and it is a ticking time bomb.

To be perfectly honest, I have seen plenty of developers like this and, unless they are just out of school, they won't change. I say cut your lossless now, its only going to get harder to fire him as he continues to spew out more unmaintainable code :)

Share Improve this answer Follow

edited Feb 10, 2009 at 14:56

community wiki
2 revs
Maurice Flanagan

---

2

It's very unlikely that management will get rid of him if he is really bright.

The whole project may be closed, of course, but there will be no use in CVS and documentation then, anyway.

No management will fire a good programmer only to hire a bad one.

Tell him that it will help **him** to get rid of management whenever he wants to.

What is he wants to change job? He can tell management: "OK, people, everything's just like you asked me: checked in, documented and under you control. I'm done with my part, I pack and leave".

Share Improve this answer Follow

answered Feb 10, 2009 at 13:41

community wiki
Quassnoi

2

Can the team be successful with out him? If so push the issue and refuse to accept any code that isn’t properly documented or doesn’t meet other standards. Hopefully this will get the point across but it might just make him angry and cause him to quit. If the team can’t be successful with out him then you’re out of luck until you can train a replacement up to his skill level which may not be worth the time and effort.

Share Improve this answer Follow

answered Feb 10, 2009 at 13:44

community wiki
Jared

If the team can't survive without him, you already have a problem. You always have to think about the "hit by a bus" scenario. – Kibbee Feb 10, 2009 at 16:35

▲

2

▼

+1 to ocdecio - if he is a star developer, then his code should be based on such a high-quality design that it documents itself.

Having said that, the frustration could be that although he's excellent in technically-demanding areas which are interesting to him, he doesn't muck-in with the delivery of features - only you will know whether this is a problem for your organisation.

Having a "guru" available can be an absolute life-saver - or at least it used to be, or has StackOverflow made that role redundant?

Share Improve this answer Follow

answered Feb 10, 2009 at 13:44

community wiki
Duncan

Code can never truly be self-documenting. The code only tells you what it *is* doing, not what the developer *intended* for it to do. Useful comments describe the author's state of mind when they wrote the code - their intent, why they're doing it this way, etc. – 17 of 26 Feb 10, 2009 at 14:19

I agree, but I don't think that code commenting is the answer. As I mentioned in another comment, I like to have documentation which goes alongside the code which describes it at a more conceptual level. Having said that, comments certainly don't hurt (as long as they are kept up-to-date!) – Duncan Feb 10, 2009 at 20:52

2

Don't let the code be released until it has passed code review and only allow it to pass if there are enough comments and/or documentation for the code which he has written for the current feature/project.

**Edit** Bring it up in his appraisal. Documentation / commenting code can be given to him for his "Areas of improvement".

:-)

Share Improve this answer Follow

answered Feb 10, 2009 at 13:49

community wiki
WestDiscGolf

2

You could also add automated quality checks that would prevent him checking-in his code until it was sufficiently documented.

That is if you can persuade him to check-in in the first place! (Which is ESSENTIAL, imo)

Share Improve this answer

Follow

answered Feb 10, 2009 at 13:50

community wiki
Duncan

2

There's a lot of folks here on the "no comments, so what?" bandwagon here. Literate code without comments is perfectly possible, but just because someone is smart and doesn't comment doesn't necessarily mean that they are writing literate code.

So let's clarify. You already said he doesn't document his code at all, either in comments or separate documents, and he doesn't use any source control. How about:

- Is his code understandable despite the lack of comments?
- Does your team use any kind of issue tracking (e.g. FogBugz, Bugzilla, etc) which he participates in?
- Is his code under test?
- Is there anyone else on the team who actually is at least somewhat familiar with how his code works?
- Is he willing to at least acknowledge that he could stand to make some changes in the way he works with the rest of the team?

If the answer to all of these questions is "no", you have a big problem. Just being smart doesn't necessarily make someone an asset. Do you have any guarantee that he won't leave your company tomorrow, or get hit by a bus? How screwed would you be if that happened? Is it worth the risk?

Share Improve this answer Follow

answered Feb 10, 2009 at 14:25

community wiki
Adam Bellaire

---

▲

2

▼

I think this is pretty typical in any environment. How do you get someone to do what you want? This is exactly what "How to win friends and influence people" is all about. Dale Carnegie was not about manipulation, but managing people.

It sounds to me like he's just inexperienced and needs some experience and guidance.

Do you think you can sit down and talk to him about these issues? Telling someone they're doing something wrong often seems like the wrong thing to do (especially in today's western society where we don't want to hurt other people's feelings) but I think you can get very far by calmly and honestly explaining the issues and talking them through. It helps if the person respects you and your opinion, which is a whole other issue and is talked about in the book mentioned above. Make sure he understands that these are serious issues. Also, that in any future development jobs he'll be expected to do these things too so it's a good idea to practice them now.

I don't think waiting until the next performance review is a good idea. Piling up a bunch of negative feedback and delivering it all at once is just a bad idea and I really disliked it when that was done to me.

Share Improve this answer Follow

answered Feb 10, 2009 at 14:29

community wiki
Sarel Botha

Often this type of person has Asperger's or is borderline autistic, and so needs to be spoken with in a very open and direct manner, because social cues don't get through.
– Mike Dunlavey Feb 10, 2009 at 17:06

2

Code documentation is over-rated. CVS training is easy.

A good class should reveal its purpose through its methods and properties.

Documenting the model outside of the application is also easier to flow and comprehend.

I would bring it to his attention, if you can't get it resolved looks like you will be losing a star developer.

Edit: Oops - used CSV instead of CVS, to many imports and i use svn heh.

Share Improve this answer Follow

edited Feb 10, 2009 at 14:50

community wiki
2 revs
Syntax

Definitely agree with documentation outside of code... I find a high-level, English description of the system in question, with passing references made to the classes within it, serves a much better purpose. – Duncan Feb 10, 2009 at 13:46

@Duncan: As long as the "high-level" remains as such. I've seen dozens of design documents filled with too much detail that were simply ignored by the developers. – Dan C. Feb 10, 2009 at 13:55

@Syntax: Code documentation might seem over-rated while you're working on your own code (or at least in a project you developed). A maintenance programmer that's supposed to fix a bug 5 years after you left the company might think otherwise :) – Dan C. Feb 10, 2009 at 13:57

When I write a component I expect to be re-used within my company, I write a little "How To" tutorial which guides the developer around the public interface. As long as I get time that is! – Duncan Feb 10, 2009 at 14:01

@Dan C. I use an integrated document system for code to declare the basic purpose of a class, however I still think that good code shouldn't be so convoluted that its confusing to anyone. A nicely planned model should have documentation before it even reaches the phase that coding starts. – Syntax Feb 10, 2009 at 14:52

1

1. Get him to use automated execution-verification tools. (See my answer at "How to ask questions to an obsructionist?")
2. If he overcomplicates and does not use SCC, he is *not* an excellent developer -- these things are important parts of software engineering.
3. In the very unlikely event that he has some irreplaceable brilliance in areas like algorithms, assign him to work in that area, e.g., defining algorithms, and get a real programmer to do the coding.
4. Use static analysis code to understand and clean up his code.

Share Improve this answer Follow

edited May 23, 2017 at 11:53

community wiki
2 revs
Joshua Fox

1

Pair programming. Find hima pair that will "complete" him for requirements you just listed. You will solve a problem will source control, documentation, question every his action, etc. You also train other guy by using first guy strength

Share Improve this answer Follow

answered Feb 10, 2009 at 14:52

community wiki
LicenseQ

---

1

From what you describe, this fellow is distinctly **not** a star developer. Programming is a team sport, and those who do not play well with others do not add much value to a project.

Personally, I might not remember code I wrote 6 months or more ago, and very much value a history of the changes in some sort of Source Control.

If you had regular code reviews with this guy I think you would see that he's not as stellar of a developer as you think he is.

Share Improve this answer Follow

answered Feb 10, 2009 at 15:03

community wiki
Chris Ballance

1

I agree with most people on this thread. One way to put him on the hot spot is to have Team Code Reviews.

For the first code review meeting choose code from a team member that is open and will accept the recommendations. The "star" developer will have the chance to see how the code review works and then you can schedule his code to be reviewed next. Give him some time to prepare for the next meeting and by then he should be at least commenting his code.

The intention of code reviews is not to put shame on people but rather to collaboratively identify issues and places for improvement, but it will be a good way to put the star developer on the hot seat.

Share Improve this answer Follow

answered Feb 10, 2009 at 15:17

community wiki
Handerson

1

In my opinion, you need to drop this guy. It sounds like his code is unreadable and he is definitely not a team, nor a safe, player.

It is also a very bad practice to allow someone to see themselves as indispensible. If you let him keep this up, his practices will get worse, not better. When he leaves, for everyone does, you will be left with an enormous headache of tangled code. You need to cut your losses now if he will not shape up.

Finally, keeping this developer without reigning him in sets a poor example from junior developers. If you force them to work correctly, you risk resentment from the "why him and not me" crowd. If you don't, you get a bunch of hacks who work like your "star."

In short, unless he shapes up very, very quickly, it's time to drop him, for the health and sanity of your entire development staff.

Share Improve this answer Follow

answered Feb 10, 2009 at 16:25

community wiki
Tom Moseley

---

1

We had a similar problem when I started my current job a little over 3 years ago, our lead developer was a cowboy. He was really smart but very erratic, some of his stuff was very well documented, but so overly complicated it was imposable to maintain, or code first and figure out what he was building later, he would throw out code to see what would stick.

He left about 2 1/2 years ago, and it's still haunting us when we find some of his old code, and in his defense that's how the development shop here was run back then, in the last 3 years I have been on a crusade to change that mentality.

When you become a star developer it stops being so much about how well you know the system, code, framework, architecture, etc. and more about

1. Writing elegant, maintainable, flexible, readable code
2. Working with the rest of the team: using source control, leading by example, etc.
3. Documenting what each method, class, object, etc is doing
4. Researching new and better ways to do things for yourself, and your team

if your not following all four of these at some level your not a star developer, more so if you refuse to try/learn to use them then it's time that that should find other employment opportunities one way or another, I hear starving artist is popular with this time of person (the whole misunderstand genius thing)

Share Improve this answer Follow

answered Feb 10, 2009 at 16:48

community wiki
Bob The Janitor

1 2 Next