

Is Test Driven Development good for a starter? [closed]

Asked 16 years, 2 months ago Modified 12 years, 4 months ago

Viewed 2k 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 12 years ago.

Expanding this question on how I learnt to pass from [problem description to code](#) Two people mentioned TDD.

Would it be good for a starter to get into TDD (and avoid bad habits in the future ?) Or would it be too complex for a stage when understand what a programming language is?

tdd

Share

edited May 23, 2017 at 11:57

Improve this question

Follow



Community Bot

1 • 1

asked Sep 26, 2008 at 3:37



OscarRyz

199k • 119 • 396 • 573

14 Answers

Sorted by:

Highest score (default)



10



TDD is meant to be simpler than the "traditional" method (of not testing it till the end) - because the tests clarify what you understand of the problem. If you actually didn't have a clear idea of what the problem was, writing tests is quite hard.



So for a beginner, writing tests gets the thinking juice going in the right direction, which is contractual behaviour, not implementation behaviour.



Share Improve this answer

Follow

answered Sep 26, 2008 at 3:39



Chii

14.8k • 3 • 39 • 45



4



I wish TDD were around when I was first learning to program, and that I had picked it up before getting so entrenched in the 'old way' such that it's very difficult for me to learn TDD...



Share Improve this answer

Follow

answered Sep 26, 2008 at 3:40



Erik Forbes

35.8k • 28 • 101 • 122



-
- 2 I agree. I consider myself a fairly experienced programmer but I am finding the TDD learning curve to be unbelievably steep. It's easy if you're writing a math function library but testing real-world business problems is where TDD seems to come unstuck. – [Damien](#) Mar 12, 2010 at 6:22
-



Experiencing TDD Rules All

4



I also think that ideally TDD would be very helpful in the early stages of learning. In hindsight I know it would of helped me approach the problems in a completely different light.



What I'm perplexed about is that when one is learning, there are so many new concepts being absorbed that confusion can start to set in very early. Therefore, while I do think TDD would be super helpful, I don't think it can be something that's learned successfully by one's self.

Just like anything else in life we tend to learn best when *somebody is physically teaching* us. Showing us how they approach the problems in a TDD manner can do so much more than reading about it in books or on the web. I mean, this can't hurt but it's not a substitute for a mentor that can truly show you the ropes.

Experiencing TDD is everything so if you can have somebody teach you how to TDD during those early

stages, I think learning as a whole would be accelerated beyond what anyone would expect.

Share Improve this answer

edited Jun 20, 2020 at 9:12

Follow



Community Bot

1 ● 1

answered Sep 26, 2008 at 5:06



Scott Saad

18.3k ● 11 ● 66 ● 85



3



```
def self.learn_tdd_and_programming_together?  
  if you_have_tdd_mentor_sitting_next_to_you?  
    "go for it"  
  else  
    if language.ruby?  
      "it's possible, there is quite a bit of good  
stuff out  
      there that could give you a chance of  
learning programming  
      with TDD from the start. It's sort of in the  
ruby culture"  
    elsif language.dot_net?  
      "learn TDD after you learn the basics of  
.NET"  
    end  
  end  
end
```

Share Improve this answer

answered Sep 26, 2008 at 3:48

Follow



ryw

9,635 ● 5 ● 29 ● 34



1



it's certainly a lot to take in, but having said that I wish I started out writing unit tests. What would actually have been good was if I had a mentor at my workplace who could have guided my TDD progress. I've been self learning TDD on and off for about a year and there's a lot to cover and the more you do it the more involved it gets, but it's really starting to pay off now for me.

Share Improve this answer

answered Sep 26, 2008 at 3:41

Follow



[lomaxx](#)

116k ● 58 ● 147 ● 180



1



I think this [comment](#) illustrates that it can be a very good thing for beginners to learn straight up.

Share Improve this answer

edited May 23, 2017 at 11:57

Follow



[Community Bot](#)

1 ● 1

answered Sep 26, 2008 at 3:42



[Geoff](#)

3,769 ● 2 ● 30 ● 24



1



My programming motto is:

1. Make it run -- the program solves the problem
2. Make it right -- the program is designed cleanly and there is a small amount of duplication
3. Make it fast -- optimized (if needed)



Test Driven Development handles the first two.



I think a beginner should be taught TDD so that he knows how to make programs run. IMHO, only then can good design techniques be taught.

Share Improve this answer

answered Sep 26, 2008 at 3:58

Follow



jop

84k ● 10 ● 57 ● 52



1



I think yes. Studies even found that the **benefits are largest for beginners**. It gives you more guidance for writing the code. You know what the results and behavior should be, and write the tests. Then you write the code. **Tests pass. You're done.** And you know you're done.



Share Improve this answer

answered Sep 26, 2008 at 7:03



Follow



Paweł Hajdan

18.5k ● 9 ● 52 ● 65

2 I'm not sure if I agree. I think it can often take an enormous amount of knowledge and skill to be able to describe a problem as a test. – [Damien](#) Mar 12, 2010 at 6:24



0



Yes! Definitely.

Share Improve this answer

answered Sep 26, 2008 at 3:39

Follow



Andrew Stein

13.1k ● 5 ● 38 ● 44



This does not provide an answer to the question. To critique or request clarification from an author, leave a comment below their post. – [phs](#) Aug 30, 2012 at 19:32

@phs Actually, it does answer the question... Would it be good for a starter to get into TDD? . It's just not a good question for Stack Overflow. – [saluce](#) Aug 31, 2012 at 14:31



0



I think it's not good for someone just learning programming. How will that person know what to assert?
:P TDD is for design, not for testing. Once a person knows how to program, it'll be a good thing to start studying the TDD approach.



Share Improve this answer

answered Sep 26, 2008 at 3:40



Follow



[cruizer](#)

6,151 ● 2 ● 29 ● 34



0

First you need to understand how to code well. Read, study and practice that until you have a good handle on it. Once you have that, look into test driven design - it's very powerful.



Share Improve this answer

answered Sep 26, 2008 at 3:40



Follow



wprl

25.4k ● 11 ● 57 ● 70



0

An important benefit of TDD is defining doneness. In simple algorithmic programming, if you come up with a couple scenarios where correctness is easily asserted, it's easy to enumerate them in a unit test and keep coding until they all work.



Sometimes unit testing can be hard for beginners, if there are many dependencies and you start to run into scenarios where mocking objects is necessary.



However, if you can make a simple statement about correctness, and it is easy to type out, then definitely write it down in code.

You may also note that if a simple statement of correctness is not easily described, you may not fully understand your problem.

Good luck...

Share Improve this answer

answered Sep 26, 2008 at 3:45

Follow



Nathan Feger

19.5k ● 11 ● 64 ● 73



0



It really depends on your definition of a "starter". If by "starter" you mean someone with absolutely no programming background, then no, I don't think TDD is a very good way to start out. A programmer needs to learn the basics (avoiding infinite loops, memory allocation, etc.) before worrying about refactoring and test driven development.

Share Improve this answer

answered Sep 26, 2008 at 3:50

Follow



Kevin Pang

41.4k ● 38 ● 122 ● 173

Yes, someone that attempts to open Microsoft Word to write code. – [OscarRyz](#) Sep 26, 2008 at 3:59



0



code is code whether it is the thing you're trying to spike out, or a test.

Learning TDD at the very beginning has a lot of value. It's one of those skills that should be a habit. There are a lot of us out there that understand and like the value of tdd but years of programming have instilled some some habits that can be hard to break later on.

As far as TDD being for contract design/code implementation/testing it's all of those things. Will TDD bring you to the perfect code? No, experience and

studying the craft will help you mature your coding approaches. But TDD is a very important tool for every developer.

The use of TDD will hopefully help bring you to a design that is testable. And a design that is testable is in theory well encapsulated and should adhere to the open closed principal.

In my opinion as long as people view TDD as something that's a niche tool or is somehow optional while writing code, those people obviously don't get the value of TDD.

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

answered Sep 26, 2008 at 4:49

[Follow](#)



Casey MacPherson
