

Any Behaviour-Driven Development Tools for PHP?


Behavior-Driven Development (BDD) is an evolution of Test-Driven Development (TDD) which pushes the focus away from the concept of testing to one of specifying behaviours in a form more easily understood. It's actually pretty hard to explain BDD in a few short lines but I'll throw in a quick intro after posing a question:

Are there any usable PHP Behavior-Driven Development libraries? Anywhere? I get the feeling PHP has once more fallen behind Ruby and Java which have four such libraries between them. Anyway - if not I'll have to cheat with SimpleTest for now.

Anyway, in the past I undertook to learn Ruby and got started with the RSpec BDD library after a few weeks - thanks to Brian D. for the introduction to RSpec. If you prefer a Java outlook, there's JBehave or JDave. The reason for having a replacement for the traditional Unit Testing tools is obvious after you've practiced BDD for a while. Unit Testing tools don't support the BDD terminology and ways of thinking - it's possible to cheat with JUnit (and PHPUnit/SimpleTest in my case) using test messages and careful method naming but the result isn't very pretty even if its functional, and you lose a lot of the benefits by having to put up with the "test" oriented nature of an xUnit lib and its lack of a fluid language base.

Here's a typical RSpec behaviour spec of a non-existent Fleet class for a game. Just as with TDD, we write the specification (not "test!") first and then write enough code to meet it.

```
describe Fleet, "when first instantiated" do
  before(:each) do
    # just like using setUp() in PHP UT
    @fleet = Fleet.new
  end
  it "should contain no Ships after formation" do
    @fleet.population.should eql(0)
  end
  after(:each) do
    @fleet = nil
  end
end
```

BDD avoids any mention of "test"; since it revolves around specifications. I know that sounds a bit vague - just trust me that BDD does not involve Unit Tests in the classical sense. RSpec notes itself as providing a "Domain Specific Language with which you can express executable examples of the expected behaviour of a system". See, not tests and no mention of units either 


This tiny example can be summed up by followed the keywords:

```
Describe a Fleet when first instantiated:
- it should contain 0 Ships after formation
- ...
```

This small text snippet "describes" a Behaviour by setting out what "it should" do. In fact a good BDD library will generate such text summaries for you. Honestly, after a few months of occasional RSpec usage this has become a pretty intuitive way to think of things - more so than TDD which requires you hit "behaviour"

keyword on your own initiative.

It gets cooler when you create a Squadron which needs to behave like a Fleet. RSpec gets around this by introducing the concept of "shared behaviors" which I will not describe here except to say it's a really neat approach involving Refactoring.

Any other BDD fans in the audience? Any last ditch memory of a PHP BDD library? 


Posted by Pádraic Brady in PHP General, PHP Security at 21:28

I hadn't heard about BDD before, but it sounds quite cool, for as much as I've understood of it. Anonymous on Sep 19 2007, 22:24

Any unit testing framework can do BDD: it's just the same thing with a different vocabulary - "TDD done right". A new label implies that TDD and BDD are two different things - ironic given that the aim was to avoid confusion (the many meanings of "test") by changing the language. Anonymous on Sep 20 2007, 01:46

I have to partially disagree - the different vocabulary of BDD is one of the reasons I actually like it. It removes a certain level of confusion and focuses on a terminology far more suited to the task. If I dug a random Devnetwork post I wrote about TDD there's a good chance I'd have used the magic "behaviour" word before. Having a language that's accessible is, in my opinion, extremely important.

In this, a traditional UT framework invariably fails. Even though I do agree you can BDD using one - I call it "cheating" for want of a better term - I certainly notice a difficulty in translating a behaviour (not tests of outcomes) to a set of tests. Again, it's a language barrier - something the mind isn't quite getting or finds difficult to work with when faced with a test/assert based language and zero fluid interface.

That's why I found RSpec to be a cool thing. I think it's very easy, but very wrong, to underestimate what difference a suitable language makes. For me, BDD definitely adds something I never quite had with TDD/UT and that's really all that matters (to me ).


And really, they didn't create a new term for just a few language updates. More to BDD than the language focus. Anonymous on Sep 20 2007, 03:15


Apparently, http://blog.daveastels.com/files/BDD_Intro.pdf is the manifesto of the BDD movement. I did a quick skim, and while I agree with some of the general ideas, such as more atomic unit tests and emphasis on testing all levels of behaviors (from a single method unit to entire subsystems), I fail to see what benefits this "BDD" would create, besides the naming difference that might make it clearer how to properly organize things.

RSpec sounds like building a wrapper language for the tests. In a way, that's neat, but is it really necessary? Anonymous on Sep 20 2007, 07:23

It's a conceptual difference - the key thing is that BDD is not a huge departure, it's a relatively small evolution on existing concepts. The reason for a new library is that current libraries don't supply the updated language and tools (this is not to say such libraries are wrong or useless, merely that BDD nails work better with a slightly different hammer).

Consider it as a simple improvement - If you are doing TDD really well, then you're likely already doing a variant of BDD without realising it. The purpose of BDD (the new term) is to push doing TDD (without tests, since TDD is primarily a design methodology anyway which just happens to use Unit testing) really well, updating the language to boost accessibility, and trying to divorce TDD goals from dependency on the system it's being used to design.

I am of course completely crap at making the point of difference here in a comment . Too short a space. I can only say that after using

RSpec/JDave and getting to grips with BDD, I find it works for me. Something definitely changes when you use BDD. Which is a personal opinion of course so don't take it as Gospel 

. Anonymous on Sep 20 2007, 16:24

PÃ¡draic, if you cannot find BDD libraries and decide to "cheat" with SimpleTest, would it be possible to post a blog entry about how to do that? That (to me) would be a welcome introduction to BDD. Thanks. Anonymous on Sep 20 2007, 21:38

Paddy, wouldn't this basically be a behavior defined:

```
$widget = new GreatestWidget();  
assert('is_null($widget->name)');
```

That being the case, you could do that in .pht files and use --TEST-- to define the behavior in human-readable terms, or in PHPUnit or SimpleTest use testHumanReadableExplanation() method names to explain the behavior.

Not sure I see the reason for another test scaffolding when testing is considered so esoteric within the PHP community as it is.

That said, I do like the aim of BDD, I just think its a little misguided. Work on education of what testing is, then you'll have people realizing "no, I'm not testing, I'm writing out the specification in terms of code." I will admit that the term "test" does have a reactive feel to it that's ingrained in children from elementary school: you learn the material, then you take a test (mentioned in the specification vs verification in the above linked PDF). Anonymous on Sep 20 2007, 22:29

You are correct, and that's why BDD is an evolution, not a revolution 


. Where BDD departs from TDD is removing at least one language barrier for newcomers. If one were to create a feasible PHP API for an imaginary PHP BDD library:

```
$this->specify($widget->name)->shouldEqual(null);
```


Anyone could come up with: "Widget name should equal null". I think BDD is on to something here that's not easy to follow coming from TDD (where it's obvious, and so undervalued, since we're already experienced TDD users who already get TDD is not about testing). Does this make it easier to help a newcomer evade the TDD-is-testing trap? I think it could help. It's getting more buzz so time will tell.

>Not sure I see the reason for another test scaffolding when testing is
>considered so esoteric within the PHP community as it is.

Not a test scaffold per se though 

. Testing is everywhere in PHP, but TDD is not. In working in the Zend Framework I find myself dragged into implementation debates - and that's at the proposal stage where the last thing I want is a time consuming exchange on the merits of ArrayAccess for non-existing code. I seriously doubt one could find more than a few TDD driven components in the ZF. I'm not sure you could even find a Mock Object (possibly forgivable though given PHPUnit's history) though I really hope that blatantly false 

Part of BDD's motivations is that TDD education doesn't appear to be very successful. Yes, there are converts - but it's not yet an overwhelmingly standard practice. Somewhere, people are looking at TDD and losing the plot. Even in pair programming (I did some TDD training a while back this way) it takes a while to get someone from looking at tests as the goal, to using tests as a tool to get to the real goal - describing class behaviour. The B word needs to hold more significant placement in TDD literature since it's the one word every beginner needs to be taught early.


Not that I'm abandoning TDD wholesale 

. I just want to use BDD for a few projects in PHP and see where it goes. Since I do most work in PHP, having a PHP tool would be useful. Maybe if I have some time I'll see if it's possible to create a BDD friendly scaffold on top of SimpleTest (a bit better than long method name/assert messages alone). Anonymous on Sep 20 2007, 23:34

If I want to apply BDD in PHP - and suggest some practices then I probably will... Anonymous on Sep 20 2007, 23:46

Have you checked out PHPSpec?

<http://code.google.com/p/phpspec/> Anonymous on Jan 12 2008, 02:58

lol - Matthew, check the revision history of phpspec 

Anonymous on Feb 5 2008, 06:46

@Matthew:

I'm writing PHPSpec 

. Started it after this post went out and nobody could suggest a BDD framework in PHP - nor could Google...

Anonymous on Feb 5 2008, 06:50