Testers in Software Development [closed]

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

Viewed 1k times

11

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 10 years ago.

Improve this question

In my company we develop a software product. Until now we haven't had any testers, so basically the developers were the testers, and of course the customer and users (not good).

Our team now consists of 4 developers and we work mainly with Cruisecontrol, Flex, ASP.NET, IIS, MSSQLServer and WebORB. I am urging the managers to hire a tester but I wonder if testers are normal in software development. So:

1. Are testers necessary in product (or large scale project) development?
2. Should testers only do test work? Or can you expect from a developer or graphical designer to test half of the week?
3. Where can you find good testers (I suppose there isn't a degree in software development testing)?
4. Is it the task of the project manager of technical teamlead to test everything?

thx, Lieven Cardoen

ps: Thx, Vinay, we do have Unit Tests but indeed, Unit Tests can't cover what Testers can.

testing project product

Share

Improve this question

Follow

edited Feb 8, 2009 at 13:03

Svante

51.4k ● 11 ● 83 ● 124

asked Feb 8, 2009 at 10:36

Lieven Cardoen

25.9k ● 54 ● 158 ● 250

10 Answers

Sorted by: Highest score (default) ⬍

7

✓

1) Are testers necessary in product (or large scale project) development?

Yes. Somebody's got to take responsibility for assessing when something is tested enough and working out which bugs need fixing or can ship because there are workarounds.

2) Should testers only do test work? Of can you expect from a developer or graphical designer to test half of the week?

Testers often do customer support work or work with the customer to develop requirements. The testers can act as internal voices for the customers... and if they interact with the customer enough should feel a sense of responsibility for getting a suitable quality product out of development they know the customer will want.

3) Where can you find good testers (I suppose there isn't a degree in software development testing)?

I bet there is a degree somewhere. A lot of the testers we have are computer science university students who are doing a year out in industry before they go back for their final year at uni.

4) Is it the task of the project manager of technical teamlead to test everything?

Not necessarily. It depends how big the team is, if it's small, then yes, someone can double up and do both roles. For bigger projects though, these are different people.

Remember. Having a tester is not an excuse for developers/programmers not to test code as they write it or create unit test. Developers still have responsibility for developing a good product. They should never attempt to make up an excuse for a bug they created by blaming a tester for not finding it.

Share Improve this answer Follow

answered Feb 8, 2009 at 12:10

Scott Langham
60.3k ● 42 ● 136 ● 210

3

It's question #10 on the Joel test:

http://www.joelonsoftware.com/articles/fog0000000043.html

Should tell you everything you need to know. :)

Share Improve this answer Follow

answered Feb 8, 2009 at 11:14

Andy Hume
41.6k ● 10 ● 51 ● 59

2

It is better to have testers in the team. We, developers can't get so much of defects when we test, but if it is tested by a tester then he catches so many defects before releasing the defects.

It is better to have unit test cases also. When we develop a new feature, just we have to update the test case and run it.

Share Improve this answer Follow

answered Feb 8, 2009 at 10:43

Vinay
4,773 ● 7 ● 35 ● 43

2

Testers are necessary in both cases. Even if your development is test-driven, I think the role of the tester often focuses on the external requirements of the project - does the project deliver the expected product that will fulfill the requirements?

I find that in large corporate environments you often have a goldmine of good testers at call-centers or those who do customer service. They often have a very solid understanding of the business processes, problems and requirements.

In such a context we have often let the testers work with constructing real-life test cases in back-end systems too, sometimes these are even used by integration tests that the developers write. We've had great success with letting the developers *ask* the testers for data/scenarios to test for CI-based automated testing.

You *can* expect other roles to take part in testing. I really think everyone should be focused on quality and testing, but unfortunately *everyone* cannot be responsible.

Share Improve this answer

Follow

answered Feb 8, 2009 at 10:51

krosenvold

77k ● 33 ● 156 ● 209

2

1. From some size on, absolutely (I'd say about 10 developers).
2. Testers could also do build and integration work. In smaller groups developers have to test, because there is nobody else to do it.
3. Good question. Maybe some of your developers like testing.
4. No, especially when the project gets larger.

I've worked on a large project (hundreds of developers), in our group about fifty. Our group had an integration & testing group of two or three full time people and a bunch of students.

Share Improve this answer

Follow

answered Feb 8, 2009 at 10:52

starblue

56.7k ● 14 ● 99 ● 153

2

1. ON large scale, yes. However there are many different methods and types of testing. These include user, regression, unit, and integration testing. Try and automate as much as you can. Check out Selenium (IDE), Molydbenum, Usage Scenarios and Agile development.

2. A developer or designer should decide if they have met acceptance criteria but if they test their own work then it's kind of like writing your own exam before you sit for it. Developers testing other developer's work isn't much better in my opinion.
3. Dunno
4. I think a project manager won't have time for rigorous testing. It's really the job of a dedicated tester who knows to be diligent and can interact with the project manager.

Share Improve this answer Follow

answered Feb 8, 2009 at 11:56

Rimian

38.3k ● 17 ● 122 ● 118

2

> 1. Are testers necessary in product (or large scale project) development?
> 2. Should testers only do test work? Of can you expect from a developer or graphical designer to test half of the week?
> 3. Where can you find good testers (I suppose there isn't a degree in software development testing)?
> 4. Is it the task of the project manager of technical teamlead to test everything?

1 - on product development when you have > 10 customers: hell yes. ESSENTIAL. Same on a large scale project. You can skimp when you are small, but once you get over a certain size, the pain of updating (eg) 100 customers around the world outweighs the salary of even ONE tester.

2 - yes, tho there is some overlap into support work too. Developer should do basic testing - does it work? - but it's up to the testers to do the exhaustive, end-to-end, weird-use-case type testing. It's a waste of a developers time to do that, I think. Graphic designers shouldn't test - well, they user test, I hope, but thats even before it gets to the developers.

3 - some developers make good testers. some support people make good testers. other than that - just find them at random. advertise. etc. there are no degrees, but someone who is pedantic, can stick up for themselves, and knows more about how the end environment holds together than how each line of code works would be good.

4 - no. the PM holds the project together, and co-ordinates the testers, devs etc. the tech lead should be, ya know, leading the tech team. not testing.

Obviously, there is leakage between the roles. Sometimes, EVERYONE should be doing some testing, but this is more to get maximum coverage just before RTM, not on a day to day or week to week basis.

Unit tests are a great start, as they catch the logic errors, but they can't be expected to catch crazy user interactions, or problems that only show up after your app has been running for 72 hours+ - unit tests will NEVER EVER catch these. Your customer will, but then you will not have customers for long :)

BTW, I've "been there, done that". I've tested on customers and had proper testers at different stages (startup vrs bought by a big company) of the same product. The product was a LOT more solid once we had testers, and customers were happier too (plus, it's hard to roll out a small critical patch to 400 sites around the world - catch it before it ships!)

Share Improve this answer Follow

answered Feb 8, 2009 at 12:32

Nic Wise

8,119 ● 2 ● 32 ● 30

2

Never underestimate the value of specialists.

Most people who aren't testers, particularly developers, don't enjoy testing and won't do it well. If you ask a graphic designer or a developer to spend half of their time testing, at best you will lose 50% of the output of a good designer/developer and gain 50% of a poor, expensive tester. At worst you will lose them completely because they will find somewhere better to work.

In the case of developers, they are usually too close to the code to be able to test it objectively. They will make assumptions based on their knowledge of the internals. A developer testing their own code is particularly bad.

The project manager should be responsible for ensuring that everything is tested but they shouldn't be doing it themselves. They won't have enough time or the necessary expertise.

I previously worked for a consultancy company. We had no specialist testers and would instead use whichever consultants were currently without a project. None of them had any testing expertise and as a result most of them weren't very good testers. We'd get bug reports like "the system doesn't work anymore" or my personal favourite: a screenshot of an application showing how slow it was (a screenshot of the application running quickly would not have looked any different). They would abuse the bug tracking system too (or bypass it completely in favour of their own home-brew Excel spreadshhets). It was a nightmare.

Share Improve this answer Follow

answered Feb 8, 2009 at 14:08

Dan Dyer
54.4k ● 19 ● 135 ● 166

▲

1

▼

I think what testers (can) bring to the table is a fresh set of eyes on a product. I have found that I'm likely to follow the happy path when running software, whether this be via UI interaction or utilizing classes I've written. When you know how something is supposed to work, it is a bit unnatural to do things the wrong way, to exercise the product in a way that it wasn't intended or to try things in an order that nobody who knows how it works would do.

My answers to your questions are as follows:

1. Yes. Testing is a necessity and thus some sort of tester is needed. Typically, unit tests will catch a lot of the low-level problems, but usability testing and 'requirements' testing determine if it meets the claims on the glossy brochure. If you claim your software does 'X', then part of the tester's job is to make sure it really does 'X'. We've had tester uncover some problems on platforms that I don't normally use. It is good to find those problems early.
2. Maybe. We cross-test our products in-house but have a separate testing group too. We tend to (in-house) find the lion's share of the issues but the full-time testers find things sometimes that we'd likely have never found. If you split time between developing and testing, it needs to be clear that testing isn't an afterthought. If I'm overburdened working on my developing and have a hard time spending the requisite time testing your product, then I'm not going to be that effective as a tester. Time management is essential if you are going to have developers doubling as testers.
3. Not sure. Some groups (IV and V groups, for instance) in an organization do only testing. I suspect there are a number of folks out there that have no business or desire writing software but can test the

heck out of a product. Some of our best testers at my last job didn't write any code at all.

4. Depend on your project, but I'd say no, generally. Responsibility is something that is spread out at my shop. The lead isn't responsible for all testing. We are all accountable.

Anyhow, that is my two cents.

Share Improve this answer

Follow

answered Feb 8, 2009 at 14:26

itsmatt

31.4k ● 11 ● 102 ● 165

1

Everyone will say testers are required, and that's the politically correct answer. But testers are just one of many tools available in the quality assurance toolbox. You can ship perfectly fine software without testers. For example, I wonder if Stackoverflow has a test department?

A tester will **measure** the quality of the code your developers produce. But the availability of quality measurements does nothing to improve quality.

Testers don't come for free. You'll have to change development to deliver to the test team instead of the customer. This can lead to a disconnect between customers and developers: the testers might find bugs that customers couldn't care about, and ignore bugs that are very important to the customer. In addition, you'll have to create and maintain a separate testing environment.

Good testers will help you develop better, and especially release more smoothly. But your mileage will vary.

P.S. In a big company testers are essential: they enable blame shifting. Say a bug causes customer harm -> executive is upset. At this point the executive is ready to do nasty stuff to your team. With a test team, you can shift the blame to the test team, who shift the blame back to you. A compromise rolls out: you introduce new processes, the test team hires more testers, and the executive says he's personally improved stuff.

Share Improve this answer Follow

answered Feb 9, 2009 at 14:47

Andomar

238k ● 53 ● 385 ● 410