

What do testers do? [closed]

Asked 15 years, 10 months ago Modified 8 years, 7 months ago

Viewed 8k times



8



Closed. This question needs to be more [focused](#). It is not currently accepting answers.



Want to improve this question? Update the question so it focuses on one problem only by [editing this post](#).

Closed 8 years ago.

[Improve this question](#)

I work in a company where the developers QA the work of the other developers checking things such as adhering to coding standards through to whether it works or not.

Now this seems to work extremely well for us but I can't help feeling we are wasting development time on something a dedicated tester or testers could do.

The problem is I've always worked for this company so I have never worked with testers so don't know what function they have within a development team other than the mile high view of "they do testing".

We also tend to hire graduate level people so someone would have to guide them through all their tasks for a

time.

In summary, what do testers do within your company and how do they fit into your development and release processes?

testing

Share

Improve this question

Follow

edited Feb 8, 2009 at 13:24



Dan Dyer

54.4k ● 19 ● 135 ● 166

asked Feb 8, 2009 at 12:56



Garry Shutler

32.7k ● 13 ● 89 ● 120

-
- 1 It seems to me you should consider hiring somebody with experience to run your testing department, instead of a new grad and hoping he gets it right – [MattBelanger](#) Feb 8, 2009 at 13:39
-

I just noticed this very similar question also asked today:
stackoverflow.com/questions/525514/... – [Dan Dyer](#) Feb 8, 2009 at 13:49

@mabwi yes, that would be better but I'd be more likely to convince management to hire a junior. – [Garry Shutler](#) Feb 8, 2009 at 13:56

12 Answers

Sorted by:

Highest score (default)





11



Their job is plain and simple. Break the application. You always know when you have a good tester, because you're always a little annoyed when that person comes around your desk/cube. The reason for this is that you know that if the tester is in your general vicinity, they've found something wrong with what you've written. All the excuses start to pile up in your mind of 'Well, you're not using it right!', etc, but in the end, you know that the tester is right, and you've just made a mistake in your programming.

Good testers can *find* bugs. They can think like a user, to verify the business rules, etc, but they also act like a user when they click in unusual patterns to force your application to break. It may seem like they're abusing the application and using it in a way its not meant to be used, but that's their job, and that's why they're paid as testers.

You know your tester needs to be replaced when they can't find anything wrong. Believe me, in any complex system, there's always *something* wrong, and it's the tester's job to find it.

That being said, it's of utmost importance to use dedicated testing people, especially when dealing with any application that has a hefty UI component.

Share Improve this answer

answered Feb 8, 2009 at 13:05

Follow



David Morton

16.5k ● 3 ● 66 ● 74

This is a great description of their general function, but how do they go about their testing? What triggers them to check the code base? – [Garry Shutler](#) Feb 8, 2009 at 13:14

- 1 They don't check the code base. They check the operation of the application. When it doesn't do what it's supposed to do, they come and hassle you - and then you check the code base. – [ChrisA](#) Feb 8, 2009 at 13:17
-

But there must be a method to when they check the functionality of the code base. If they waiting until everything in a spec is developed then you're adding a significant period of time on to the end of the development process. How do they integrate with the process itself? – [Garry Shutler](#) Feb 8, 2009 at 13:20

- 1 They're not interested in the code base. The code could be spaghetti code for all they care, as long as it works. You check your code base in a code review with a manager or peer, not with a tester. And testers don't usually wait until the end of the development cycle to begin testing.
– [David Morton](#) Feb 8, 2009 at 13:29
-

Yes, they don't care about the code base itself but the code has to be written before they can test it so they are dependent upon the developments of the code base.
– [Garry Shutler](#) Feb 8, 2009 at 13:43



9

Following on from [David's answer](#), a good tester is worth his or her weight in gold - and good contract testers can be very expensive.



I worked with a superb tester some years ago. I was the tech lead at the time, and he was the bane of my life, but his worth was incalculable.



He was highly organised, and extremely intelligent. He wrote his own test plans, based on limited documentation of requirements and functions. Mostly he ran the application, and from his understanding of the business, worked out what it should do, and where it fell short.

His attention to detail was nothing short of awesome. Everything he reported was completely reproducible, documented, and came not just with error reports, but suggestions for alternate behaviour. This was hugely useful, of course, since not all bugs result in the application breaking.

He was also flexible enough to recognise where things were high priorities, and (temporarily!) stop hassling us over the things we didn't have time to do.

So we got UI feedback, bug reports, even suggestions for where the requirements had been misunderstood.

He worked me hard with what he found, but we had a strong recognition of our common goal, namely a high quality system. If you're out there, Nicholas, I wish you well.

To the OP, I'd suggest you look for someone with these skills.

Share Improve this answer

edited May 23, 2017 at 12:30

Follow



Community Bot

1 • 1



4



A good QA department will do several things:

1. Write a test plan according to the functional specification of the product. This helps flush out the functional spec and find areas where it needs to be improved/changed.
2. Find bugs in the product This one is obvious.
3. Test the usability of the product from a non-developer point of view. This one goes far beyond finding bugs - it doesn't do you any good to have a bug free product if no one can figure out how to use the thing.

As to how they fit into the process:

1. As soon as the development team feels that the functional spec is complete, it is given to the QA team so they can write test plans
2. When the development team has a relatively stable build with a reasonable amount of functionality, it can be given to QA so they can start looking at it. At this point, QA's focus is just to get familiar with new release and point out any glaring usability flaws rather than hammering the thing to find bugs.
3. Once the developers say "ok, I think we're ready", QA starts the bug finding mission.

4. Developers and QA work to resolve all issues. Bugs are all fixed, dropped, or postponed to future releases.
5. QA has final say on whether or not the release is let out the door

Note that 3 and 4 above can vary a lot depending on whether you're talking about a new product or a release of an existing product. If you have an existing product, and awful lot of testing can be done in parallel with development.

Share Improve this answer

edited Feb 8, 2009 at 13:39

Follow

answered Feb 8, 2009 at 13:27



17 of 26

27.4k ● 13 ● 68 ● 85



3



Ideally testers should be involved from the early on in the project, so that they can formulate the test plans. This will involve, among other things, authoring test scripts. Actual written test scripts are important for repeatable testing (e.g. for regression testing new releases). As well as testing functionality, the plans will cover testing of different platforms, testing usability and testing performance.

Testers execute the test plans and find and report bugs with enough detail to minimise the developer's work in

fixing the bugs. This means taking the time to figure out exactly how to reproduce issues. Testers generally cost less than developers, so it's better for the company if the testers are doing this than if it is left to the developers. Testers tend to be better at it too since they don't make the assumptions that developers do.

Testers shouldn't really be getting into the realm of checking adherence to coding standards - this is better left to automated tools. Testers don't ever need to see the source code.

When you have good testers working full-time on a project, they quickly become the experts on the requirements (more so than developers who are only working on part of the system).

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

[edited Feb 8, 2009 at 13:41](#)

[Follow](#)

answered Feb 8, 2009 at 13:34



[Dan Dyer](#)

54.4k ● 19 ● 135 ● 166



2



Programmers test code, testers test applications. Testers read the specification, think of scenarios that could cause problems (what if two people do this at the same time?) etc.

They then document a series of tests, try them out, report the outcome, and so on.



Share Improve this answer

answered Feb 8, 2009 at 13:06



Follow



Peter Morris

23.1k ● 12 ● 93 ● 163



2

See Joel's [Top Five \(Wrong\) Reasons You Don't Have Testers](#) for a description of what testers do and why they are good for software companies.



Share Improve this answer

answered Feb 8, 2009 at 13:13

Follow



Yuval F

20.6k ● 5 ● 45 ● 69



This is the thing, I kind of know that we need at least one tester but I have no idea of what to get or expect them to do. I need to be able to do more than say "go test stuff".

– Garry Shutler Feb 8, 2009 at 13:18



2

Developer tasks:

- From inside outwards - focus on code
- Assertions - verify data flow and structures
- Debugger - verify code flow and data
- Unit testing - verify each function
- Integration testing - verify sub-systems
- System testing - verify functionality



Tester tasks:

- From outside in - focus on features
- Scenarios - verify real-world situations
- Global tests - verify feasible inputs
- Regression tests - verify defects stay fixed
- Code coverage - testing untouched code
- Compatibility - with previous releases
- Looking for quirks and rough edges

Share Improve this answer

answered Feb 8, 2009 at 14:13

Follow



[HTTP 410](#)

17.6k ● 14 ● 79 ● 129



1



Traditionally, in a large IT services company, a tester's role tends to vary slightly with the nature of the development process adopted. Traditional waterfall or iterative projects tend to involve testers designing test plans, writing test cases and clarifying requirements during that process, executing them (both manual and automated) and clearing the app for production moves. They also regression-test other apps that could be potentially impacted. For most part, they do not ever look at code but in some special cases, they do validate database entries, especially in scenarios where batch jobs or other legacy systems are involved.

Agile projects on the other hand are increasingly causing a blurring between the responsibilities of a tester and a developer. With frameworks like Rails or Django, the developer has a much better view of the "big picture" than ever before, so it generally does not make sense to have a large, dedicated, purely testing team. And with a perpetual beta philosophy, a good part of the testing is done by actual end users. So a much leaner, more dev-savvy testing team tends to help Agile projects (at least inside enterprises). Amongst other things, it helps when testers can put together scripts to automate regular test cases instead of having to rely on expensive tools (like Win/Loadrunner)

On an average, a tester's motivation levels tend to be lower than developers. At least in my organization, a lot of testers want to "grow up" to be developers, although some of them do understand that becoming a QA/Assurance consultant is a career in its own right.

Share Improve this answer

answered Feb 8, 2009 at 14:33

Follow

community wiki
[krishashok](#)



0

You should hire people to do the testing.

Testers use the application and report bugs they find. If you have a spec, they can test the application against it



to report any inconsistencies.



No product release can have quality if it's not tested.



Share Improve this answer

answered Feb 8, 2009 at 13:02

Follow



[luiscubal](#)

25.1k ● 9 ● 59 ● 84

We do this at the moment but through developers but I don't see how this could fully occupy a tester's time if this is all they do. – [Garry Shutler](#) Feb 8, 2009 at 13:06

Any time they spend not-testing they spend reading the spec's and thinking of problems they can test later. They also document a lot. – [Peter Morris](#) Feb 8, 2009 at 13:08



0



Actually recently I've come to realize how to tell a bad tester from a good one. When the tasks are closed because no bugs are found and an hour later you yourself crash the application because you felt like "It's stupid but what will happen if make that kind of input?" and did it, that is a good sign that someone (tester) didn't do their job.

I regularly report bugs somewhere in our software and all the time I fell like "That's not me who should be doing this".

Share Improve this answer

answered Feb 8, 2009 at 13:30

Follow



[User](#)

30.9k ● 22 ● 81 ● 108



0

To get a better handle on this I highly recommend Gerry Weinberg's book "Perfect Software: And Other Illusions about Testing" ([sanitised Amazon link](#)).



It's full of excellent insights that will make you think about testing in entirely new ways.



HTH



cheers,

Rob

Share Improve this answer

answered Feb 8, 2009 at 13:44

Follow



Rob Wells

37k ● 13 ● 84 ● 147



0

In contrast to your scenario, I have been working closely with the testers. I find them very helpful for the fact that they understand well where my software fits in the over all scheme of things. They know better about the application with which mine interacts. Their inputs in that respect are very valuable.



Share Improve this answer

answered Feb 8, 2009 at 13:46

Follow



Prabhu. S

1,793 ● 2 ● 21 ● 34