ASTQB™

AMERICAN SOFTWARE TESTING QUALIFICATIONS BOARD

Finding and Retaining Skilled Software Testers

Read on to learn:

▶ Why you need good testers.

▶ What makes a good tester.

▶ How you can get these testers.

▶ How you can retain these testers.

People in our industry seem to fall into three categories:

1 Those who cannot attract skilled testers to their organization

2 Those who can get the right folks, but can't keep them

3 Those who have the right people and those people stay

For those of us in the first two categories, we naturally resent the guys in the 3rd category. But, aside from the letting our resentment fester, we need to take a look at why some organizations do so well with attracting and keeping skilled testers while others continually struggle.

Let's start with some fundamental questions. Why do you need good testers to produce a quality product? How do you get those people? And, finally, what can we do to retain the good ones once we finally find them?

Why do we need good testers?

Anybody can test, right? Sure, but if you care about the results of that testing, you understand that not everyone will produce the same results. And testing isn't limited to using the keyboard to work with the software. In order to produce a quality product, testing must be a part of an overall quality approach. Good testers drive good testing throughout the organization. They help keep everyone focused on quality. You can get testers who will test the final product that the developers toss over the fence, but that won't build a quality product. Remember, testing alone doesn't enhance the quality of a product – it has to be built in.

How do good testers keep the focus on quality?

They do it by making sure there are quality practices, and testing, throughout the development life-cycle. This means the developers need to build in quality and check it with unit tests (preferably automated) and peer reviews. Let me digress right here and hop on my soap box. Peer reviews, if done by a true peer, are only going to catch mistakes and practices that the developer should have caught. Peer reviews should be done by someone with stronger skills than the developer so the developer will learn and grow as a result of the review. OK, back off the soap box.

When our developers are already thinking about quality by testing their code as they develop it (or even building the tests first with a test-driven design model), we are already well on the road to a quality product. What do testers have to do with this? Testers should be part of the code review. Testers may not be able to read the code, but they can certainly listen to what is discussed and ask pertinent questions such as "How will the software handle this situation?". By participating in the code review, the tester leaves with an understanding of the code and maybe a bit of insight into what parts of the code are more complicated and may require more testing. Traditionally though, testers do not attend code reviews. They usually aren't programmers. That doesn't matter. Testers can be significant contributors to a code review but, even if they can't actively contribute, they can listen and learn.

What about those unit tests? Should testers review them? Ideally, yes. This only works if the unit tests are recorded somewhere, perhaps in a unit test automation tool. It is a good practice for the tester to walk through the unit tests with the developer, both to understand what is being covered and to advise on areas that might also be targeted. Remember, the goal here is to build a quality product as a team.

Similarly, having the testers involved in integration test planning is also beneficial. This requires an understanding of the integration points (which will be useful knowledge when it comes to system testing) and perhaps a technical understanding of APIs and how to test them. If testers are not technical enough to build API tests on their own, they may be able to leverage tests built by the developers. Working together on this task builds the team and keeps the quality focus.

What makes a good tester?

The ability to apply good testing techniques is critical for creating a quality product but only educated and dedicated testers will know the techniques and will be able to select the proper situation in which to apply those techniques. Testers with an ISTQB certification will know the fundamentals of good testing, although you will certainly want to be sure they know how to apply those fundamentals in your organization. The ability to work well with the developers and build the relationship needed to retain a quality focus throughout the project requires strong personal skills.

Clearly we need strong testing skills, technical knowledge and good personal skills. But we also need flexibility and a willingness to drive forward even when there are many obstacles in the way. Ideally we wouldn't need these skills. Ideally the projects would flow in a steady pace and our processes would always work and there would be complete harmony in the organization. Ideally. But how many of us work in an ideal environment? Do you get perfect requirements and perfect time frames? Adequate headcount? Do you ever change, or want to change, methodologies?

Good testers who will stay with a job are flexible, adventurous and love a challenge. They're willing to dig down and figure out the requirements. They don't play the victim or whine. They get on with the job and do the best they can and look for ways to improve. There are the people who will make your organization successful. These are the people who will stay with the job because they enjoy it. And these are the people who will help you build and deliver better products — products that can stand the test of use in the real world.

So how do we get these wonderful testers?

Good testers are out there. And some of them work for companies that aren't smart enough to retain them. You can benefit from that. First we need to attract them with a good job description. Job descriptions can say a lot about your organization.

Take a look at this excerpt from a job description for a test manager:

> The Test Manager is a professional tester, who can deliver continuous integration and delivery. Passion for testing as an art, exceptional communication skills and lean thinking is expected of this person to deliver and meet requirements, whilst building a testing team that are also passionate about delivering world class results. Approximately 30% of this role will focus on technical skills, resourcefulness, and innovation.

> It is an expectation of this role that the Test Manager remains technically relevant, and has the ability and willingness to be hands on and help with testing when required as well as being comfortable communicating complex data to enable the senior management team to make informed and supported decisions.

These folks know what they're talking about and they're looking for someone to join their organization that understands what the job really is. I was impressed with this description and applied for the job (sadly they re-organized and closed the job opening). Just from what I read there, I was excited about the prospect of joining such a knowledgeable company.

Contrast that with this:

▸ Lead, support other tester(s) with technical assistance and mentoring:

▸ Provide day to day guidance and support to other tester(s)

▸ Assist in ensuring the appropriate flow of testing workload

▸ Ensure regular updating of testing knowledge to other tester(s)

▸ Assist in personal development needs of other tester(s)

▸ Guide and assist less experienced tester(s) and quality check their work

This reads like a generic job description. It describes the work, but it doesn't exactly grab your attention and make you want to update your resume.

So the job description matters because casual job seekers won't follow up if the job doesn't sound interesting. Desperate people will still apply, but do you want those? I always look for the note of realism as well. Clearly the first job wants you to remain hands on. I'm fine with that. The second job has a bit of a warning in it. I will need to "assist in personal development needs of other tester(s)". uh-oh. Am I joining a band of social misfits?

Our good job description should net us some excellent candidates. Now we need to review resumes (check for typos! And make sure the person really has an interest in testing) and set up the interviews. When looking for good people, it helps to understand what makes a good candidate. What does your environment need? Independent thinkers? Well-behaved sheep? People that will sit still and run through their test cases day after day? People who can deal with horrible requirements and ever-changing priorities? Be realistic. There's no point in luring someone into a job they will hate. So when you're interviewing, be honest about the good and bad points of the job. Good testers will know when you're lying – we're trained to ferret out the truth!

Also be sure that your interview questions are going to get you the right people. Interview for attitude (a good one), confidence, organizational skills, maturity, empathy and a sense of humor. People without a sense of humor may be difficult to work with and may take things a bit too personally. You don't want

them to have the skin of a rhinoceros, but you also don't want them to come running to you with hurt feelings when the developer has said something less than polite. Look for information that is volunteered. I know I'm interviewing a real tester when they get a glint in their eye and say to me "I love a challenge". That's someone who is in it for the long run. They won't be deterred by lousy requirements, exploding software and grumpy customers. They'll take that as a challenge to do better, do more, and over deliver.

A good exam is also a nice way to weed out those who just don't have the right mind frame to be a good tester. This exam needs to be given to all candidates, in order to be fair, and graded according to published grading criteria (to avoid favoritism).

Once you've found the person you want, hire them for a fair wage. You can lowball someone's offer, but let's face it, that's just starting off the relationship dishonestly. Give them an offer that is fair and they will respect you for it, even if they wanted more money (and who doesn't!). If you haven't got enough money to make a market compatible offer, be sure to emphasize the other strengths of your organization. Maybe you can offer better job security, advancement opportunities, training, or a great work environment. There's more to a job than just money and this is the time to emphasize your organization's strengths.

Some people will say, "This is all good information, but we can't afford to get the right people and retain them." Really? Can you afford not to? In reality, bad software is expensive. Very expensive. Sometimes catastrophically expensive. Good software testers are one of the ways to invest in the future of your organization. Take the time to build with quality and spend a lot less time fixing the problems. Anyone with a basic knowledge of house constructions knows this is true.

Once you've got the right people, how do you retain them?

It starts with good management. Surveys have repeatedly shown that the primary reason people change jobs is because they are unhappy with their manager. A minority will change jobs for other reasons that are likely beyond your control. So what do people want from their manager? Number 1 is honesty. Number 2 is consistency. An honest manager is trusted and while people may not like what they have to say, at least they feel that they are not being misled. It is critical that testers be honest – that's what they get paid for. It's even more critical that the test manager deal honestly with the testers and other managers. Credibility counts.

Consistency is an interesting thing. People feel more secure if you are consistent, even if you are consistently bad. They understand what you are doing and you are predictable. If you are not consistent, no one knows what to expect and they are generally uncomfortable and will probably seek to avoid dealing with you. I've had a few managers in my time that were wildly inconsistent. You never knew if you were going to be praised or abused for your work (or even praised and abused for the same work in the same day!).

People will stay in an organization where they feel they are valued. You need to care about them, personally and professionally. When you think of the employment agreement as a long-term relationship, you will understand the importance of caring about

people. If they need some time off to pick up the keys to their very first house, give them the time off! Don't make them file for vacation time. Celebrate with them. If they have a sick kid, let them attend to it. Let them work from home when it makes sense. Make allowances for employees to be human and to have lives outside of work.

Have you ever worked for a manager who has no clue what they are doing? I have. It gets old very quickly. A good manager is technically competent and savvy enough to be able to understand schedule commitments and roadblocks and the various bumps along the way. As a manager, it's not easy to stay technical. It requires time and an investment in yourself. It also requires the confidence to be able to ask your people to teach you new stuff. You have to stay current with technology and you have to be willing to learn. Pretending you're an authority will quickly lose their respect and once that's gone you have an uphill climb to get someone to work happily for you.

Let's face it though – you will make mistakes. Recognize that and admit it when you do and correct it when you can. I've made a few scheduling blunders in my time and that have resulted in my team having to work extra hours. When that happens, you have to bring them into the problem, explain how it happened and ask for solutions. Be willing to bear some of the burden as well. If I messed up the schedule, then I should be working weekends right along with everyone else.

While you're doing all these things to keep everyone in your group happy and working, don't forget to just be yourself. You'll do that a lot better than pretending to be something you are not. I worked for a manager who was very shy. He hated talking in front of people and preferred to hide in his office. His wife told him he had to get out more, but that clearly made him uncomfortable

and didn't make a good impression. When he finally stood in front of the company and admitted that he absolutely hated public speaking, everyone had a much better impression of him and made an effort to seek him out. It hadn't occurred to people that he was just shy; they thought he was arrogant and didn't want to mingle with the underlings. One honest admission from him made a huge difference. Even better, he prefaced his speech by explaining that his wife made him do it. Very human. And very honest.

Some organizations seem to work hard to dissuade people from staying. Poor raises, poor promotion opportunities, a less that good working environment, etc. Yet, people will stay with companies that have these issues because they like the work and they like their manager. So keep the work interesting. Provide learning and growth opportunities. Make sure people are as happy as is reasonably possible. You'll find your retention problems will diminish in direct proportion to your investment in your people. Work with your management to build a good environment. And above all, try to be the manager you would want to work for. People will stay. They will also tell their colleagues that your organization is a good one. That will solve your recruiting problems too!

Software testing is a relatively small community. Software quality is critical to the success of an organization and we can all cite headlines from organizations that have failed to meet the quality challenges. You need good processes and effective methodologies to build quality products. But, most importantly, you need to find, keep and grow good people. People are a long-term investment. If you want to consistently produce high quality products, you need to have high quality people. So go out there and get some – and be sure you build an environment that will retain them.

Learn how ISTQB Software Testing Certification can help you recruit and retain strong testers.

ISTQB provides a professional pathway and benchmark that builds and keeps high standards of quality in your organization.

That means fewer defects, faster turnaround time, and greater cost savings.

Learn more now at www.astqb.org

or contact us at info@astqb.org

ASTQB™

AMERICAN SOFTWARE TESTING QUALIFICATIONS BOARD