

How to Increase Test Team Success

Successful testers need to be continually critical of other people's work. Yet this critical approach can spill over into other aspects of our work. Therein lies the paradox within every test team. How do we prevent that continual criticality from denting our own motivation and leaving the test team dispirited? In this article we look to the brighter side of testing.

Traq Software Ltd
St. Mary's Court, The Broadway,
Old Amersham, Buckinghamshire.
HP7 0UT United Kingdom
Telephone: +44 (0)1494 582037
www.TestManagement.com

How do we define Test Team Success?

I'm lucky to be working with an extremely talented test team, yet I get the feeling that we haven't quite reached our true potential. So what's preventing us from reaching it? It's not that we don't have the right people or the right skills. It's not even the test process or the project. What holds us and every test team back are issues unique to testing.

The American Heritage Dictionary defines testing as "a procedure for critical evaluation," so it stands to reason that a huge part of our job involves being critical. It's probably also fair to say that being a tester means we need to be skilled at thinking critically. Does it hold true, though, that the more critical the test team is, the more successful it will be?

Defining success for a test team is always difficult. We will always release software with bugs. We will always wonder if our test coverage was as thorough as it could have been. Yet if we are to accept these as part of our job, how do we know if we've been successful? These issues can make it difficult to create a successful test team.

Constantly Critical

It can be difficult and de-motivating to be part of a team that is constantly critical. Test teams by their very nature are more critical than complimentary. A tester has to look for all the negative aspects of the software. The problem for many testers--and I'm no exception--is letting this pessimistic, critical attitude creep beyond the software you are testing.

Take, for example, a great test team I worked with a couple of years ago. We were working with a company that had decided to relocate down the road to a fantastic new building on the edge of a nature reserve with all of the facilities you could ever want. On paper this move looked fantastic. In fact, in reality it turned out to be fantastic. But reality wasn't going to stop this test team from pulling the idea to pieces. Out of all of the teams being relocated, the test team was by far the most critical about the move.

This is where the paradox lies. If your test team is positive, upbeat, and happy, you can almost guarantee that you've got the wrong people. You need a team to be critical, judgmental, and, well, negative. To be a good tester you've got to be pessimistic ("I know bugs are still in there somewhere"). You've got to be negative ("I'm not sure dev has implemented that functionality quite as well as they could have done"). If your testers are saying things like "I'm sure there aren't any bugs left," then you aren't dealing with a successful test team.

I've worked with a lot of talented testers, many on par with some of the most talented developers I've known. But when these testers lift their heads from the software, they can find it very difficult to remove the critical hat. They criticize project management, development, managers, tea ladies--you name it, they've got something critical to say about it. You can't blame them. They need to be critical. They're good at being critical. I mean, they're paid to be critical.

Perception of Success

The development team has one clear goal: to create a piece of shippable software. It may want to add just one more bell or whistle before shipping, but in general the goal is achieved once the coding stops and the product ships. For testers things are much different--it comes down to the question of when enough is enough.

Our goal is to find every conceivable bug in a piece of software, even if someone says it isn't feasible. While development may meet its goal, we as testers are always thinking, "We haven't quite finished our task yet." Consequently, we're always left thinking we haven't met our goal.

That being said, I have worked with a number of testers who perceive testing as a positive role. While they know that they can't find every bug, they also know that increasing the number of bugs found contributes greatly to the overall quality of the product.

Real Success

It can be difficult--almost impossible in fact--to define exactly what makes a test team successful. Is a winning test team more critical? Does it achieve 100 percent test coverage in a product before it ships? Realistically, a successful test team should aspire to the following ideals:

- A team ethos with a critical focus yet positive outlook
- A clear understanding that goal setting for testers needs to be unique

To help achieve these ideals, adopt the four attitudes presented below:

1. Highlight the good points within your team or project.
2. Set realistic, clear, and achievable goals for both the team and yourself.
3. Encourage developers to be positive about the testers' efforts.
4. Look for positives in the software.

1. Highlight the Good Points

It's easy to maintain a critical attitude. Conversely, when you put a little bit of thought into it, it's easy to come up with positive comments about the team, the project, or the test process. Make an effort to comment on one or two positive points every day.

This goes for everyone! Not just the test manager. The responsibility for engendering a positive team attitude falls to everyone. Put yourself in the position of the average test manager, who might have twelve testers to manage. It takes a lot of effort to lift the spirits of twelve people who highlight the negative points day in and day out. So help yourself, your manager, and the rest of your team by highlighting positive points daily.

I recently had a test manager pull the whole test team into an impromptu meeting to discuss a potential stop-ship defect that surfaced (a situation which we all find ourselves in at some point). You have several choices in such a situation: bemoan the potential defect, or, more constructively, recognize that bringing everyone together to discuss the issue really helps strengthen the morale of the team.

2. Set Clear Goals

Setting clear goals is crucial for both the team and the individual, including yourself. Testing sometimes seems to go on forever, and a team that feels there's no end in sight is likely to end up despondent. Define test areas, but make sure they have boundaries with clear completion criteria.

We're all familiar with the principle that priority 1 and 2 defects need to be fixed before shipping a product. Why not apply similar principles to your test cases? Prioritize the test cases, and state clearly the need to complete all priority 1 and 2 test cases as a team. This gives the testers something to aim for and feel good about when they achieve it. Most off-the-shelf test case management tools, as well as open source tools like QaTraq, allow you to prioritize test cases.

3. Encourage Developers

The development team and individual developers can help as well. It must be difficult, as a developer, to continually have other people pick holes in your work. However, some of the best and most talented developers I've worked with have always been positive and grateful for the testers' efforts.

Developers with the strength of mind to encourage and support testers in this scenario really help to maintain a tester's enthusiasm. If nothing else, it's in the developers' interest to help maintain the testers' enthusiasm, as it is likely to lead to more defects found before a product is released.

4. Look for Positives in the Software

Finding bugs is good from the tester's perspective, even though a bug report by its nature highlights the bad. The test process encourages a negative mindset. Instead of focusing on the negatives, why not emphasize the positives?

How many of us, when pushed to provide a status report, jump straight to the number of defects we've found in the software so far? Why not demonstrate how 95 percent of test cases have passed for a certain piece of functionality? Again, many off-the-shelf and open source test management tools allow you to focus on results for specific areas of functionality. If development is nearing completion on a certain aspect of functionality and you've completed a test run over this functionality, highlight how well testing has gone. You'll be surprised how your spirits lift when you take the time to compliment others for something they've done well.

Conclusion

In many ways, the challenges facing test teams are different from those facing other types of teams. The need to be constantly critical and the open-ended nature of testing present far greater psychological and technical demands on the tester than many realize. To address these challenges and improve a test team's performance, a far greater emphasis needs to be placed on positivity in testing. While testing may be predominantly critical, don't overlook the importance of being positive!