

**HOW TO KEEP & MOTIVATE
YOUR QA TEAM
--WITHOUT INCREASING YOUR
BUDGET**

*Marina Gil Santamaria
March 2008*

INTRODUCTION

So you went through the process of pre-scanning CVs, reading long and longer resumes, interviewing and more interviewing, meeting, deciding, and finally hiring your team of QA professional engineers. Congratulations! Now, you can sit and “relax”, right? Wrong!

The job market is against you...It is healthy and growing in North America & EMEA, and booming overseas.

The job market is against you, and looking favorable for your QA Engineers. Let me give you a couple of examples. I just checked some online job boards and there are 607 QA jobs in Boston on Monster, 150 in NYC according to Dice, and 400 openings in San Francisco as per LinkedIn, even as off-shoring projects continue to take off. What does this mean for you? Well, if I can easily see so many opportunities out there, so can your team. And as a manager you should be aware that employees usually leave a manager, not a company. So the first thing to think about is...Are you being a good manager? And by good I really mean things like: are you coaching your team while empowering your team members? Are you working with them to plan their professional growth? Are you truly showing deep appreciation for their hard work? Are you letting your team balance work and family life via flexible schedules?

Job satisfaction among QA professionals has been traditionally lower when compared with their development counterparts and other IT departments.

Asking yourself these questions are (or should be) “standard” management philosophy, but as a QA Manager or as a QA Director, there is an additional factor that you should consider. Job satisfaction among QA professionals has been traditionally low when compared with their development peers and with those in other departments. Why? External misconceptions that are out there such as “anybody can do QA”, “hire some out of school kids to test our applications”, or “QA folks are in reality ‘developer wannabees’, can really have an impact on your team’s morale. Well, I am quite sure that you have heard those comments or stereotypes before...and guess what, so have your QA members. Hmm, no wonder that sometimes QA teams seem de-motivated, discouraged or disengaged. Who wouldn’t be?

The rest of this article will take a closer look at some common QA misperceptions, reason why they are wrong, and gives you some specific follow-up action items that you could act upon to keep and maintain a motivated staff.

Myth #1: Anybody can do QA

External misconceptions that are out there such as “anybody can do QA”, “hire some out of school kids to test our applications”, or “QA folks are in reality ‘developer wannabees’, can really have an impact on your team’s morale.

Wrong. Testing is a skilled activity that requires the ability to think, explore and follow logic while questioning and reasoning at the same time. It is based on the philosophy of performing a technical investigation of a product, to provide information and report back to various stakeholders throughout the organization. And to achieve the end result of communicating back your findings, you will need to use various types of infrastructure and experimentation, logic, models, mathematical probabilities and supporting tools to build the appropriate Test Case scenarios. And don’t forget that QA teams usually have very limited product documentation (if any), and have to work in a very time-constrained schedule. Sorry, you just can’t take anybody off the street to do QA. It takes a skilled, intelligent professional who understands the needs of the organization and is ready for the task!

Myth #2: Any out of school kid can test our applications

Wrong. Your applications are a critical company asset. Depending on the nature of your business, direct profit and revenue can be impacted if an application goes down or performs poorly. And of course, a company’s reputation can also be seriously damaged when things don’t “work” right. Think about it. Will you hire an inexperienced, out of college kid to take care of your critical investments and financial assets? If you answer yes, then I would say sure, go ahead and hire an out of school kid to test your applications. Otherwise, please hire professional, experienced, skilled testers or QA Engineers (with or without certification, that is another debate point in itself) to build your QA team.

Myth #3: A QA Engineer is a “Developer wannabe”

Have you ever thought if those QA-centric miss-perceptions were present in your environment?

Wrong. While it is true that some QA engineers like to get into the code, and enjoy using automated scripts, there are other folks that prefer a “thinking & building test cases while executing manual tests” type of approach. Probably within your team you will even have a mixture of skill-sets and different personality types. Similarly you will have folks that started in QA, landed in QA from a different career, are considering a career change, or want to remain in QA for the long term. What is important to remember is this - QA is a valuable professional field that requires intellectual efforts similar to those required to develop a product. Think about it. A QA Engineer is developing the plan that will ensure the success of a company’s product in the marketplace.

How are you communicating the value that your team provides for your organization?

Myth #4: QA doesn't provide much value to the organization

Wrong. QA represents the heterogeneous users of the products that your company produces, so they are there to improve product quality, ensure a better user experience and reduce support costs. Let's take a quick look at some of the responsibilities that QA has within an organization. QA engineers are in charge of finding and reporting bugs (from critical problems to minor enhancements or documentation changes), and verify that bugs are fixed (and that nothing else broke because of code changes). In addition they typically work very closely with development, product management, customers and other internal folks to understand product functionality and use cases, as well as build appropriate test plans. They are the ones you want to vouch for the adequacy of your user documentation while it is being finalized. To top it all, QA teams assess product conformance to specifications and standards, while checking interoperability with infrastructure, complementary products and/or third party vendor products. Lastly, they have the final saying of when a Beta and/or GA product can be shipped to customers, so they can block or veto a premature product release. Impressive, right? To say that QA doesn't provide much value would be gross misrepresentation of the truth.

Myth #5: QA is a boring, repetitive job with no creativeness involved

Well, this one could be partially true, depending on the internal processes that you have in place. For example, if QA is not involved with a project from the beginning, there is not much room to bring new ideas or feedback to product's requirements. And this could translate into lack of empowerment, and not feeling appreciated. Another scenario could be if you have no automation in place, and there are truly repetitive tasks and regression testing steps that could in fact be automated in your environment. (I am not talking here about 100% automation either, but about finding out the right balance that makes sense for your organization based on your applications, environment and objectives). And, of course, boredom could set in within your team if you have internal processes that require hundreds of pages to document simple test cases, or if internal policies are based on a "write all that you do" type of approach. You need to be sensitive to this and avoid creating unnecessary procedural quagmire

SO, WHAT NEXT? HOW DO YOU MOTIVATE.

Now that you have read about common miss-perceptions, let's take a look at some pointers, which will not impact your budget, but could help you increase job satisfaction across your team. And make you a manager that your team will rally around!

Quick tips to remember:

- **Evaluate internal atmosphere**
 - **Brag about your team to stakeholders and peers**
 - **Improve relations with the development team**
- **Evaluate internal atmosphere.** Take a moment to ask yourself questions and gauge internal perceptions. Are development folks respectful to QA engineers, or do they look down at them? Is my QA team systematically involved in project planning meetings and architectural discussions? Is my department showing appreciation to my team? Once you have a better perception of how QA is internally perceived, you could identify one or two internal processes or attitudes that you could target for improvement.
 - **Communicate, communicate and communicate with your stakeholders.** Take the time to understand what your stakeholders are being measured on, and report back first on those metrics that are important to them, including business metrics. A good idea could be to come up with a list of periodic achievements from your team, and get in the habit of talking about them whenever you are interacting with your peers. For example, you could create key reports for weekly distribution (like number of bugs and associated savings, etc), and the next time you have a departmental meeting, share them and “brag” about them. The idea is to make your teams members the stars, don’t worry it will reflect great on you - while showing value to the organization.
 - **Improve QA-development relations.** If you have noticed that this relationship is not that great, think of ways to improve it. Let me give you a quick example. On my first jobs as a QA Engineer we use to get code builds at 4:30pm on Friday very frequently. Developers were happy to dump their job on us and go home to a relaxing weekend, while we were “morally obliged” to stay late and start running tests throughout the weekend. After all, they just gave us a kit, so we felt we needed to test it. How could QA morale be greatly increased in this case? Very easy, with a simple change of policy - a standard delivery date was set for Wednesday at lunch time, exactly in the middle of the week for both teams. This really helped boost team morale.

Another area to explore is departmental reporting structure. You might not be able to change this, but you could examine your dynamics, and identify the roadblocks and obstacles that could be hindering communication in your case. For example, if a development manager and a QA manager that are not communicating well are reporting to the same layer, escalating discrepancies to the top (which might work in other environments when both teams are independent) is not the right approach here. In this case you could try building a personal relationship first with your counterpart, and try to understand the other person point of view when discussing issues. In addition, you could think about having team building exercises for QA and development teams together, or having daily cross-functional meetings in your department.

Quick tips to remember:

- **Enhance your QA job descriptions**
- **Involve the team in your development cycles from the beginning**

- **Enhance your QA job descriptions.** Modify the job description of your postings to first attract better candidates, and secondly send a positive message to your QA department. What would you rather see and apply to? “We are looking for a QA Engineer to perform hands-on testing of our SDK. The QA Engineer will provide functional and boundary testing and timely feedback. They are also responsible for the project bug database and associated maintenance tasks, etc” or “We are looking for an innovative and resourceful QA Engineer – aka problem solver—who is able to conduct software testing, and bug report assessments with integrity and honesty; ability to manage own projects; good interpersonal skills, etc”. Taking the time to put together a more enticing job description, sends a strong message that your company values and appreciates their QA resources.

- **Involve QA in your development cycles from the beginning.** This is critical to ensure that QA Engineers fully understand the product, user personas and testing scenarios, enabling them to bring valuable questions to cross-functional meetings, and have an impact on design and architectural decisions right from the beginning. If possible you could start exploring Agile and Scrum technologies as a way to change your development processes. In case you are not too familiar, Agile methods are built on the philosophy of minimizing risk by developing software in shorter timeframes, which usually last one to four weeks. In theory, each of these cycles is like a mini project on its own, because it includes all the tasks necessary to release the additional increment of new functionality. At the end of each cycle, the team reevaluates project priorities together, inviting customers as well, and in theory the product should be ready to go GA at that point. Scrum is an agile method for project management, based on a peer to peer commitment throughout the project. Scrum projects are characterized by an all-hands kick-off team meeting to start working on requirements, or a release back-log file, daily team meetings and by encouraging verbal communication across all team members and less formal written documentation. And because the entire team meets everyday (development, QA, product management, etc) there are more opportunities for collaboration and more view points towards a particular task, which translates into new and more efficient ways of testing, and more automated scripts that cover realistic scopes.

Quick tips to remember:

- **Look for ways to automate**
- **Consider rotating projects and tasks across your team**
- **Involve your team in customer interactions**

- **Look for ways to automate.** If you are doing little or no automation today, think about taking the time needed to improve in this area. While it will translate in greater efficiency (and it will make you look really good with upper management), it will also be very valuable from your team perspective –it can free them from performing repetitive and boring tasks, while letting them learn and gain valuable skills that will keep them challenged. In addition, because automated scripts can be run before, during or after hours, automation can help your team increase test cases coverage, and cover a broader Test Plan without slowing down your product development cycle. From a functional or regression perspective the percentage of automation that can be achieved will mainly depend on the nature of the product that your team is tasked with testing, along with your environment, business objectives and team skills. Keep in mind that

- **Consider rotating projects and tasks.** Another area to explore is the possibility of rotating products and projects. By exposing the team to various products and tasks you could maintain a challenging and creative environment, while increasing flexibility and productivity, and building additional skills. In addition, this practice could encourage emergence of best practices, the testing of additional scenarios, and finding more bugs as “new eyes” come to each project. However, some folks might work better when they “own” specific products due to ownerships motivation. As you will find that dynamics are very different across QA teams, is very important that you take the time to talk to your team members one by one and truly heard their feedback before you decide to implement this time of approach in your organization.
- **Involve your team members in customer interactions.** If there is an interest within the team look for opportunities to engage your team in customer interactions (customer visits, demos, conference calls, booth duties at conferences, etc). It will translate into increased job satisfaction among your team, and more job security (hey, outsourced teams can’t spend time meeting customers face to face because of geographical locations and time zones differences). At the same time a better understanding of the existing customer base and prospects (who is using/may use the product, why and how) will translate into more comprehensive test plans and more realistic testing scenarios, and overall better products delivered into the marketplace.
- **Be a “good” manager.** As referenced in the opening, employees leave many times because of their managers, and not because the actual job on itself. The inverse is true also – employees will stay with an organization because of their manager. So the bottom-line here is: try to keep the working environment fun and challenging; show appreciation to your team members, and consider allowing working from home some times, telecommuting or flexible schedules. After all, as long as tasks are done in time, do you really care at what time, or from which location were they done?

The next time that you are wondering about your team satisfaction, ask yourself the following question: Am I being a good manager?

CONCLUSION

The next time that you are wondering about your team satisfaction, ask yourself the following question: Am I being a good manager? What am I doing to combat the top 5 QA-centric miss-perceptions? I truly think that there are lots of opportunities in this new era to help you nurture, keep and motivate your QA Team, so please take note and set yourself a couple of quick follow-up action items, and act on them. Your team will really appreciate it, and guess what? They will stick with you and your organization for the long term!



White Paper Title
[March] 2008
Author: Marina Gil Santamaria

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2008, Oracle and/or its affiliates. All rights reserved.
This document is provided for information purposes only and the contents hereof are subject to change without notice.
This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. 0408