



**The Art of
Test Team Management and
Motivation**

Black Hats United!

Discussion Paper

Prepared for: STANZ 2007

Author: Sarah Rees

Contact: SarahR@softed.com

Date: August 2007

Background

There is information aplenty on how to manage the test process itself, but the less predictable area of managing the team is often neglected.

The saying “Management is management is management” is true; good management skills go a long way, but managing and motivating a test team does present some rather particular problems.

When it comes to a test team, a real understanding of the personalities of testers and the challenges of working in testing will go a lot further than relying on a standard management approach.

What’s so different about a test team?

The people

A good tester thinks about software differently to a developer. We look at software and expect to find fault. In the words of Edward Kit “We focus on failure because that improves our chance of finding it”. Having this pessimistic concept of software is essential for a tester to be effective.

Good testers like to find defects! Effective testing requires skill and creativity, and brings real intellectual satisfaction.

You are managing a group of individuals who intrinsically, and by training, look for fault. Don’t expect them to restrict this to the software!

The environment they work in

Testing teams are highly dependent on others to achieve test goals. We need developers to deliver code and fixes, system and database administrators to implement the changes, business people to triage defects found and a host of others depending on the complexity of the project. This is a source of much frustration to the team, particularly when things are not going well or deadlines are tight.

There is some truth in the belief that the information coming out of testing is undervalued. Testing exit criteria are used as a “pass mark” for a project, instead of a means of evaluating risks; the information provided by testing can be played down, or dismissed as just negative. This can impact on the team’s perception of the value they contribute to the project and the organisation.

How testing is perceived by others in I.T.

A high quality system may earn accolades for the development team. If there's a quality problem, the question too often is "why wasn't it tested?"

It's hardly surprising that the average tester feels that they will get blamed when there are defects in the live environment. To make things worse, testers often feel that they are solely accountable for the quality of the system; many are still under the delusion that if only they were allowed enough time and resource, they really could find all the defects.

The expectation that testing can absorb development overruns, without a change in the scope of what has to be done, is a constant and universal problem for testers.

The perception of testing as a low skilled, easy activity is a major morale-buster.

Strategies for managing a test team

1. Set your team reasonable objectives

No test team will find all the defects. The reality is that even an organisation like NASA that adopts a 1:6 developer–tester ratio, tests over extended periods still manages to leave major defects in their code. If NASA with its extensive resources and expertise can't achieve zero defects, there is little chance for those of us trying to achieve the same with limited resource and even more limited time.

In general, people are motivated by achievable targets; Realign your tester's (and your own) expectations about what they can really achieve.

If the duration of testing is reduced, consider reducing the scope of testing; don't just rely on the old fallback of working longer hours. This sets an expectation that the team will always agree to work long hours, and reinforces the concept that testing is a low skill activity that can be squeezed into a shorter timeframe without any impact on the quality of the task.

2. Influence the business and/or project management expectation of the role of testing

The concept that testing is there to “make sure it works” is regrettably still commonplace.

Part of the role of a test manager is to sell the real value of testing to the key stakeholders, and to communicate the impact of reduced timeframes and resources.

3. Guide your team to align with the project objectives

The reality is that systems do go live without successful completion of testing, with defects outstanding, even where the test team is convinced it should not. The tester’s role is to assess the quality of the system and to provide information for risk-based decision making by the project and/or business team.

However, if testers believe they will be blamed for the quality of the software, they tend to develop a risk-averse approach, adopting a position of quality guardians, becoming rather myopic in their view of the world: They stop taking business drivers into account.

When the system does get implemented, testers may feel that their efforts have been wasted, in that the information they provide is not considered valuable enough to influence the final go-live decision. Highly motivated professionals get switched off.

I like to set the test objectives clearly in the test plan. If we have hard dates to meet due to business imperatives, I make sure that the team is aware of this from the start, as it helps them keep focussed on the project objectives as well as the quality issues.

It’s much easier to get testers to accept this position if they are not worried about getting the blame. Be sure your testers understand that the ownership of the risk will pass back to the decision makers.

4. Create a positive culture within your team – celebrate the skills that make testers good testers; don’t apologise for it

Testers have a love/hate feeling about their intrinsic skills. It’s easy to see ourselves through the developer’s eyes, and interpret ourselves in a negative way.

For me, one of the real pleasures of being a trainer has been re-setting testers' view of themselves. Many are still surprised to learn that not only is it natural for testers to be professional pessimists, but it is something to be actively encouraged.

5. Nurture a collaborative approach with the development team

The essential skills of a tester complement that of a developer. Too often though, particularly in the developer-centric world of IT, the tester is seen as deviant to a developer norm.

The scene is set for misunderstanding and conflict. It is too easy for testers to see developers as cavalier in their approach to quality, and for developers to see testers as overly critical, nit-pickers.

It's tempting to use the inherent tension between the development and test teams as a motivator; after all, uniting in the face of a common foe is a stratagem that has been used many times in the past! Although this approach might lift the morale of the team in the short term, over time it will demotivate them.

People don't like working from a position of constant conflict; it creates emotional overheads and wears them down. If it goes on long enough, three things will happen:

- 1. They leave in search of a quieter life.***
- 2. They sacrifice their beliefs and standards to remove the conflict.***
- 3. They realise that such a black and white view of the world is naïve, in which case your credibility as a manager is impacted.***

6. Reward ingenuity in your testers

It's a common mistake to try to motivate a test team by offering incentives for the number of defects they find. This usually results in the testers focussed on finding volumes of defects, no matter how trivial, rather than finding the defects that count. This trivialises their job, reinforces development ideas about the nit-picking tendencies of testers, and can initiate a "them and us" attitude in both teams.

A better approach is to review defects found by your team, and identify those that demonstrate the real skill of a tester – either in terms of the thought that has gone in to creating the test or in terms of how difficult the defect would have been to find.

7. Keep it real

In testing, we can sometimes end up dealing with abstracts; e.g. total defects found, percentage of severity defects. It's easy to lose the relevance of what we do in this context.

One approach is to ask testers to document the potential consequences of a selected defect they have found, had it been not detected during testing. This approach reminds testers of the value they bring to the project and the business.

8. "Chicken Licken"¹ complex

Whilst the professional pessimist is a trait to be nurtured in our test analysts, it does have its down side. Testers can easily slide into total pessimism about the project as a whole.

From a project perspective, this approach can lead to valuable data generated by testing becoming blanketed under a wave of negativity, and dismissed accordingly.

From a test team perspective, the difficulty becomes one of motivation. How do you keep people committed and energetic about the task in hand when they already believe it will fail?

One of the keys to this is to evidence progress. During team meetings ask each member for their rating (on a scale of 1-10) on the quality of the application. In the early days, the rating may be rather low, but you will be able to demonstrate an improvement over time. If rampant negativity breaks out, remind them of how things have changed.

The next major influencing factor is you! Your team must see that despite deadlines, you continue to champion the quality issues.

¹ For those not familiar with Aesop's fable, the tale of Chicken Licken has become the epitome of a hysterical or mistaken belief that disaster is imminent.

9. Have fun!

Don't save up all the fun for special occasions, outside of working hours. The best teams learn to incorporate fun into their working lives without compromising progress or efficiency. Testers, being particularly inclined towards negativity, need a fun environment to create balance in their working lives.

A fun environment, where people respect and like the others they work with will keep people in a role despite the stresses of the job. Too often we allow a pressure cooker environment at work, then offer cash or other incentives to people to work harder. In the short term, this might be effective, but long term, it doesn't work.

Create an environment that your testers enjoy and take pride in.

10. Encourage diversity

An effective test team is a group of individuals with their own perspective on the world. You need this.

11. Improve your estimation skills

Any good test manager should be evaluating past estimates and refining them for future test projects. If you are routinely getting the team to work extra hours and weekends to meet project schedules, you run the risk of burn out and higher attrition.

Even where your team are not paid for additional work, you should know how many hours your people are working. You may have met that last scheduled release date, but if you did so by working the team at 150%, and you do not re-estimate, you will condemn them to a working life of constantly trying to work at this rate.

Tired testers make mistakes, and miss defects. Good testers like finding defects and are not happy when their ability to do so is negatively impacted by long hours and weekend work. Reducing down test timeframes further supports the notion that testing is a low priority activity and this goes directly to the perceived value of what the testers do. Morale is impacted, and resentments towards development are nurtured.

12. Bring on the newbies!

Good managers manage succession. Very few people now remain in the same job at the same company for any extended period of time. For test analysts, the only way for them to expand their skills and their experience is to test on a variety of platforms, with different business processes and different testing challenges. If you work for a software vendor, you get this variety. If you don't, then you are more likely to change jobs frequently.

As a manager, you need to do more than get an effective team together, you need to maintain it.

Given the constraint on skilled resource availability, it makes more sense to develop the skills within your own team, by mentoring and training people to take the place of those who move on.

13. Encourage pragmatism

We work within constraints, just like everyone else in IT, and in the end, we do the best we can within those constraints. The reality is that if we have assessed the system correctly, and given accurate and appropriate information to the project sponsor, our job is done. If the system goes live despite our concerns, so be it. Our job ends when we hand the information over.

A more pragmatic stance from your testers can only be achieved once the expectations of the project/business have been managed, and a clear understanding reached as to what testing can and cannot achieve.

14. Lead from the front

The problem with a group of people, who are robust enough to stand their ground over defects, is that this will never be a team who will blindly follow! They will argue it out with you if they don't agree.

If your background is not in testing, get some training! To effectively manage a test team, you have to speak the same language, understand the pressures and be able to intelligently discuss any issues the team have identified.

Testers often feel like the poor relations. If you think you can manage a test team without a good understanding of the test process and its challenges, you reinforce the idea that testing is a lower level skill.

15. Work with your team

Morale problems need to be “nipped in the bud”, so early warning on morale issues are important. Sitting in with your team helps you to notice warning signs in time to act.

If the team are working evenings and/or weekends, work with them.

Everyone working together to meet an objective is a very powerful tool for building a strong cohesive team.

16. If you want flexibility, give it

As deadlines approach, testers can put in long hours. If you need people to be flexible enough to achieve a deadline by working evenings and weekends, then offer flexibility in return.

Once the pressure is off, let people ease off the gas pedal, work shorter hours and take time out. Flexibility must be a two way street.

17. Let testers own their role

If a tester finds a defect, let them champion it. Allow them to liaise with the developer, and agree on the resolution. This is an integral part of the test analyst role. If you can't do this for practical reasons, ensure you discuss any resolution with the tester.

Allowing testers to fully commit to champion quality issues is a major factor for job satisfaction.

In closing

The suggestions above have been developed through trial, error and observation over the past 18 years. There is no magic bullet for managing a test team, but hopefully, the ideas above will help when dealing with some of the issues that so frequently arise.

The most important factor for me is how much these things are under the direct influence of the Test Manager. We can't always influence the I.T world, but we have the capability to significantly influence the culture and success of our own teams.

Further reading

1. "Command Presence; Animate and engage people" - "Leadership Excellence" Volume 1 December 2005
2. "Are You Engaged?"; Seven ways to engage people - "Leadership Excellence" Volume 1 December 2005
3. "High Performance Software Testing Teams" - Len DiMaggio IBM, September 2004
4. "What Makes Teams Work?" [www.fastcompany.com/magazine/40 October 2000](http://www.fastcompany.com/magazine/40/October_2000). This contains a number of perspectives from different managers.
5. "Teamwork Principles" - www.abs.uci.edu/deptes/vcabs
6. "Secrets of Successful Teams" - Chris Widener. www.insiderreports.com
7. "Teamicide Revisited" - Tom DeMarco and Timothy Lister. (Excerpt from "Peopleware: Productive Projects and Teams".
8. "How to Identify a Dysfunctional Team" - Thomas Alspaugh, UCIrvine.
9. Debugging the Development Process - Steve Maguire
10. Lessons Learnt in Software Testing; - Kaner, Bach & Pettichord.