Successful Test Management: 9 Lessons Learned

Introduction

Many test managers came to management through the technical ranks. Although they may have had plenty of testing and/or engineering training and mentoring, they frequently learn management skills the hard way, through trial and error. As a manager, you have two primary jobs: to get the best work out of the people who work for you and to create an environment that enables people to get work done (so they can do their best work). This article describes some of the lessons I have learned about each of these management jobs.

It's Always the People—Help People do Their Best Work

1. Hire the best people for the job.

I meet many managers who want to hire someone just like the last person they hired. As a test manager, you have to assess the kind of person you need now, and in the near future. You might have a department full of fabulous exploratory testers now. If you hire another exploratory tester, someone even better than anyone you currently have, is that the best use of your open position? Sometimes it is, sometimes not.

The best person for the job may be someone unlike anyone you currently have in your group. The best person may not "fit in" with how you normally do things. As a test manager, hire the best people by developing a hiring strategy, and interviewing people to see whether they will fit that strategy. That gets you the best people for the job—the people who can get the necessary work done.

2. Make time for uninterrupted conversation with each person in your group every week.

As a test manager, one of your deliverables to your management is a periodic assessment of what and how your organization is doing. You also have a deliverable to your staff, of knowing enough about what they're doing and how they're doing it to give them formal and informal performance reviews. You can't meet those deliverables if you don't know what everyone is doing.

I hold one-on-one meetings with each team member at a regular, uninterrupted time every week. (When I manage more than 12 people, I meet with some people every other week.) I set aside 30 minutes every week to talk to each person about their jobs: any issues or problems they're having; if they need my help; their performance; and how well they are meeting their goals. I generally set aside one day during the week to have one-on-ones. I set up a specific time in advance with everyone, and then I meet privately with each person. If we uncover an issue that we need to discuss in more detail, we set aside another meeting time.

Many managers say that they just don't have the time to meet with everyone on their staff once a week. In my experience, if I don't plan time with people every week, they will interrupt my work, because they have to talk to me anyway.

If you *plan* time with your staff, you can reduce the number of unplanned interruptions (both theirs and yours), and understand more of what they are doing. When you understand what your group is doing, you can be more effective at helping people set priorities, reallocate resources, replan parts of projects, remove obstacles, and so on.

3. Assume the person doing the job knows how to do the job

Because many managers started as technical people, they know what it takes to do the job their staff is now doing. Well, they *think* they know now. If you've spent the last couple of years managing, you probably don't know as much as your technical staff does, specifically about how to do the day-to-day work.

You or your predecessor hired the people in your group. Presumably, you hired these people because you thought they could do the job. If you assume each person knows how to do his or her job, you will get better results than if you assume they don't know. Although some people do their work successfully with or without your assumptions, some people's work will be affected by how they feel you think of them.

Since I assume the people know how to do their job, I give my staff their assignments, ask if they need help, and then leave them alone to get the work done (unless they ask for help). I don't mean that you shouldn't talk to them while they're doing the work; you just shouldn't interfere with them doing their work. Interference can take a variety of forms:

- If you sneak up behind them, leaning over their shoulder, and asking, "How's it going?" Even after they pick themselves off the ceiling, this will still not win you points with your staff.
- If you ask every day, or worse, every few hours, how they are doing. This looks like micro-management to your staff, which is irritating. After all, don't you have any work to do? In addition, they will assume you think they don't know how to do their job.
- If you say, "I would do it this way", when they haven't asked for advice. Inflicting help is not helpful.

If you're not sure how to know if your people are succeeding, agree with each team member on when to ask for help. Everyone, including you, should choose a metric for knowing when he or she is stuck. One of my clients has a 15-minute rule—if anyone is stuck on something for 15 minutes, they have to stop and talk to someone else about the work.

When you assign work, ask if the person understands the work to do, and if he or she has the tools to do it. Decide jointly when you will check in with each other (periodic status reports, one-on-one meetings), and stick to that schedule. If the person gets stuck, he or she should feel free to come to you for help, but if you insist on interfering, your employee will come to you for help only as a last resort.

4. Treat people the way they want to be treated, not the way you'd like to be treated

The Golden Rule "Treat others the way you'd like to be treated" is appropriate for many purely social aspects of life, but not always appropriate for work.

Effective managers learn how each member of their staff wants to be treated. Some people want specific tasks and instructions, while others prefer more general information. Some people are motivated by solving new, tough, complex problems, but others are only comfortable when dealing with things they already know how to do to.

In addition to working differently, we all prefer different types of recognition. Money is not the only method of recognition you can use to reward your staff. Some people are motivated by private thank-yous, some by public recognition, some by M&Ms or movie tickets, some by a team party. Remember that whatever motivates you will not necessarily motivate every member of your staff. Talk to your team members to learn what form of recognition each prefers to receive.

Create an Environment That Enables Good Work

5. Emphasize results, not time.

Many organizations reward people based on the amount of time they spend at a job, not the results they achieve. However, hours on the job may not correlate with productivity. If you really want to improve an organization's productivity and efficiency, consider permitting everyone to work only 40 hours per week.

One objection I often hear is that "You can't get anything done here during they day." If you manage in a high-interruption environment, decide what you can do about the interruptions. Are there meetings you can cancel? Are your team members able to organize their work so that they can be maximally productive?

When people are at work more than 40 hours per week, they start taking care of *their* business at work. They pay bills and they call the people they haven't called because they've been working.

Once you create an environment where people get work done at work, and start encouraging people to work no more than 40 hours per week, you can then reward people based on what they can accomplish in 40 hours a week. I always find that productivity goes up (because people aren't too tired to do the work, and because they're not taking care of personal business at work).

When you start observing results, not just time, it's also much easier to give accurate and appropriate performance evaluations. Are your staff completing their plans and test designs? When they develop tests, how much rework do they need to do? (If you only observe how much tests I can run, I can run the same test over and over again. You still don't know any more about the product.) Plan the work for a 40-hour week, and reward the kind of work you want done in that time.

6. Admit your mistakes.

Everyone makes mistakes. They can range from forgetting a meeting to making a crazy customer commitment. It's embarrassing to admit that you make mistakes. Many of us think that we lose the respect of our group or peers when we admit mistakes.

If you don't make mistakes very often, you can actually gain respect when you admit your errors. If you forget a meeting, and then apologize, the other person will understand you, and eventually forgive you.

Whatever you do, don't deny or ignore your mistakes. Ignoring a problem doesn't make them go away, it turns the problem into a monster. At a recent client, a manager yelled at his staff in a meeting. After the meeting, he realized that he should not have yelled at the team. He was all set to let them settle down a little, and talk to them in a few days.

I suggested he talk to them right away, before they had time to get angrier with him. He resisted at first, but then relented, and spoke to each person alone two days later. Every person said something like this to him: "I was just angry with you after the meeting. If you'd come to talk to me right away, I would have chalked it up to a bad day. But now, it's been two days. I'm still angry with you, in fact, I'm even angrier, but now I'm not sure how to trust you. I don't deserve to be yelled at. How will I know you won't do this again?"

My client wasn't altogether sure how to handle the situation. He thought he'd done the right thing by waiting, and had only made the situation worse. He decided that he would never let things wait again, and talk to his staff right away.

It took his staff a few months to fully trust him again, but my client actually increased his personal power by admitting he made a mistake. Now, he and his staff can joke about the incident, and they all call it a turning point in his awareness and capabilities as a manager.

7. Commit to projects only after asking your staff if they can do it.

You're sitting in a senior staff meeting, and your boss turns to you, and says "Can we release this project next October?" It's incredibly tempting to just say "Of course!" However, your staff will appreciate you more if you can say, "I have to think about that."

Even if you've already considered the request, and asked your staff what it will take to do that work, you still may not have enough information to commit to the request. You might be in any of these situations:

- In the moment, you might confuse this request with another request.
- There may be other implications you haven't considered, since it's no longer the same time you first considered this request.
- If you "train" your managers that you answer questions on the spot, your managers will continue to press you for an answer.

When you discuss an issue with your staff before making a commitment, you've said these things to them:

- I want to know what it will really take you to do this work.
- I'm not afraid to tell my management what it will take.

Considering the work before committing to it is one case where managing "up" is a byproduct of respecting your staff. In addition, considering your staff's input may bring you respect and loyalty from your staff.

8. Plan training time in the regular workweek.

Testing, as part of engineering is a challenging and constantly changing discipline. Since it is constantly changing, consider planning training time in the regular workweek. If you don't train your people on an ongoing basis, you will both lose.

The training can be about the projects or about technology. You can do this in a number of ways:

- Hold brown-bag lunches, where each person discusses a particular area of his or her expertise. This is especially effective in a test group where your group works on multiple projects simultaneously, mostly in ones or twos. When each person works on a different project, this helps everyone understand all of the projects your organization is working on.
- Hold periodic expository talks from each department. With any luck, each department in your organization is intimately familiar with what it does, but normally other departments don't know what the others do.
- If you have cross-functional teams in your organization, you could have each team present its project to the rest of the company, or just to your test group.
- Invite outside experts to talk about specific technology or projects. These experts could be professional consultants or speakers, or they could be a knowledgeable friend or colleague.
- If you've bought a tool and already had training, consider organizing an in-house "user group" meeting, where people can share how they use the tool with others and discuss problems, tips, and tricks. This can be particularly effective with defect tracking systems and configuration management systems.

9. Plan the testing

There's never enough time to do everything you'd like to do as a test manager. So, plan what you and your test group *can* do. Decide on your mission as a test manager. Is it to find all the Big Bad Bugs Before Ship? Is your mission to assess the state of the software? Is your mission to help the development manager do risk assessment before shipping? Your mission might be one of these, or some combination. No matter what your mission, plan the testing before you get into the crazy-making period before ship, where everyone is flat out trying to do everything. You might not have to do everything. If you plan to less than everything, and you have time, then you can figure out how to do more.

Part of planning the testing is to develop test strategies for each product or piece of the product. What's critical about this piece to test? What don't you have to test? What hardware/software combinations do you need to test? Which combinations can be not tested as thoroughly, or possibly, not tested at all?

Testing is a form of risk assessment, and you, along with the rest of the project members can decide in advance, how much risk you're willing to take by testing or not testing part of the product.

Once you've decided what to test, develop release criteria for each product. Release criteria are the objective measurements of what's *critically* important for each release. Use the release criteria to decide when to release. The "It Would Be Nice If"s are not part of release criteria. The "Customer will kill us if this isn't done right"s are part of the release criteria.

If you plan the testing, and work with the project team to develop release criteria, you don't ever have to play the gatekeeper role. You don't have to stop shipment of a release. You, and the project team, or possibly you and the project manager assess the state of the release criteria. When you've met the release criteria, you ship. If you can't meet the release criteria, be honest, and decide what to do. On all the projects I've worked on, we've had to meet the release criteria, so we kept working until we did meet the release criteria. Some of my clients have had projects where they decided at the end, they had created too-strict release criteria. They documented the changes in release criteria, explained their position to the project team, and to senior management, and then shipped the product.

Summary

Testers can make great managers. Manage your management career the way you plan and develop tests: define a strategy, both for hiring and for testing; identify how to manage your staff so they can be as productive as possible and so you can assess their work; observe your own work, make any corrections and continue.

You don't have to do everything perfectly as a test manager. You need to do enough right to help people do their best work and to provide your staff with an environment they can do their best work in.

Acknowledgements

I thank my reviewers for their valuable comments: Brian Lawrence, Rachel Silber, Karl Wiegers, Jerry Weinberg. This paper is based on a previous paper, "Successful Engineering Management: 7 Lessons Learned", presented at IEEE Electro 1999.