



## Quardev Monthly

Volume 5, Issue 11

November 2011

### Welcome!

Welcome the November edition of the Quardev Monthly!

We look forward to sharing insights and helpful information in the areas where we know a thing or two: testing, quality assurance, technical writing and documentation, project management, and consulting.

In this edition we have a feature arti-

cle from Joy Shafer, *Setting Your Team Up for Long-term Success*. Joy gives some ideas for getting your team to follow software development best practices—because they work.

Enjoy the newsletter with our compliments and please contact us with questions, comments, or article ideas.

-The Quardev Crew!!

### Setting Your Team Up for Long-Term Success

*Joy Shafer, Consulting Test Lead, Quardev, Inc.*

During twenty years as a tester and/or test manager, I've seen team after team struggle with the complex problems of delivering high-quality software on time. I've been on teams that have excelled. I've also been on teams that have failed miserably. I've observed that success or failure is closely linked with efficiency, and efficiency results from following good software development practices.

In this article am not going to cover best practices. I'm not even going to discuss good practices. I'm assuming if you're taking the time to read this that you're already familiar with these. I am also not going to introduce you to a new technique or methodology. What I'm going to discuss is how to get people to actually

follow good practices.

Ah, there's the rub! You know which practices I mean. Probably most of your team knows what they are. But for some reason, you are not following them.

### Investing for Long-Term Success

One of the most common reasons teams don't follow best practices is because it takes an investment of time and resources, at least initially. Implementing new best practices may not deliver immediate results and the goal of process improvement often falls victim to competing priorities. The decision to go after a short-term goal versus a long-term goal is often justifiable, but if your

*(Continued on page 2)*



## Setting Your Team Up, Continued

*(Continued from page 1)*

team consistently focuses on meeting short-term goals at the expense of long-term goals, you will likely find yourself in a situation where delivering high quality software on time is a fantasy.

The types of practices that tend to be sacrificed are:

- Investment in test, particularly test automation
- Software quality
- Software maintainability
- Investment in infrastructure
- Employee morale and vitality

All of these things are requirements for a high-functioning software development team, just as clean oil is a requirement for a high-functioning automobile engine. If you never take your car to be serviced, it is going to leave you stranded on the side of the road—and it will be dark, cold, rainy, and in the middle-of-nowhere—and the towing and repair bill will surely be much higher than the cost of getting the car serviced in the first place. Most people understand this and service their car, if not regularly, at least minimally. For some reason, many companies don't have the same understanding of the need to maintain and support their software development processes.

If you never focus on the things you need to do to become a high-functioning software development team, you'll find yourself shipping buggy software, late and over-budget, time and time again. How do you stop this cycle?

## Strategies

### Reward Teamwork

Unless the project is tiny, software development is very much a team effort. The best way, ultimately, to consistently deliver high quality software on time, is to have a reward structure in place that rewards teams for behavior that leads to successful releases.

This seems like common sense, however, I've seen few companies who reward team effort and not a single company that does this as a significant part of their official reward structure. Imagine if, in addition to the intrinsic rewards of working on a successful team, you also received a substantial monetary bonus every time you shipped high quality software on time? How hard would you work then?

If you're the executive sponsor of a software development group, you may be thinking, "We can't afford that!" Think again. It is extremely expensive to miss deadlines and lose customers due to poor quality software. It's likely very cost-justifiable to reward high-performance teams.

### Support the formation of high-functioning teams

Probably most of you reading this are not corporate executives. You may not have much influence on corporate policy. It's unlikely you'll get your company to rally around the radical idea of compensating successful teams. Is there anything else you can do?

Yes! Encourage your management to support team efforts, and particularly team successes, in other ways:

- Provide training and support to develop successful teams
- Hire managers with proven track re-

## Setting Your Team Up, Continued

*(Continued from page 2)*

cords of building successful teams

- Recognize successful teams at meetings
- Reward successful teams with extra days off, a team party or special function, or some other desired benefit
- Make the above activities part of your corporate culture

### Tactics

There is no one method that will work for every team and every situation. I've found the following tactics to be generally useful.

#### Know best practices

If you don't already know the best practices in software development, take some time to learn them. The reason they're called best practices is because they work. They change some over time as technology changes and new good ideas come forward, so take the time to keep up.

#### Find your allies

There are usually others on your team who are as passionate about doing things right as you are. Enlist their support in fixing the problems you find. If they've been on the team for a while, they may have excellent insight into the reasons problems persist.

#### Understand the real problem

In theory, fixing problems is a straightforward process: (1) understand the problem, (2) brainstorm a solution, (3) implement the solution, and (4) examine your results. If the results are not as desired, go back to step 1. Otherwise, celebrate your success and look for the next problem to tackle.

In practice, fixing problems can be very challenging. What seems like the obvious reason for a problem may not be the reason at all. There are often layers of issues, not the least of which are roadblocks thrown up by well-meaning but misguided people.

You need to understand what drives people to do what they do and want what they want. Only then will you be able to effectively persuade them to support change. Every single person on your team has their own personal goals, either stated or unstated. Sometimes these goals are in line with team and corporate goals, sometimes they are not. Sometimes hidden agendas of team members can wreak havoc on your project.

Any time you start making changes to the status quo, you are going to make some people unhappy—no matter that your changes are overwhelmingly positive. Understand people's motivations for resisting change so that you can effectively counter them.

#### Be polite but persistent

Often the people who are resisting your efforts to implement best practices have more power and influence than you. How do you get around that?

I've found that persistence can be very effective in encouraging team members to follow best practices. If you're too heavy-handed you may get beaten down and lose your credibility. Instead take every opportunity to educate the team and management on best practices and why they should be followed. This technique requires patience. Usually if you can get the team to adopt even one best practice to good result, they will be more willing to follow your lead on your next crusade. Be gentle but relentless.



## Setting Your Team Up, Continued

*(Continued from page 3)*

### **Measure yourself**

Most teams have some sort of measurement system in place, but they may not be measuring the right things. Measure your team on how well they are following successful software development practices. Take inventory. What are your current practices? Compare them to best practices. How far away are you and what steps do you need to take to get there?

### **Set clear goals**

This should be a natural result of measuring yourself. Don't try to fix everything at once. Figure out what is causing the most pain and fix that first. If the first solution you try doesn't work, try something else.

### **Celebrate successes**

Once you start seeing success, reward it. Brag about it at company meetings. Have a party. Give away special prizes. Tell the team how proud you are of their accomplishments. Usually success gets people motivated to go on to even greater successes.

### **Conclusion**

If your team is struggling to deliver high-quality software on time, odds are they are focusing on short-term wins rather than long-term process improvement. Figure out which best practices would make the most difference for your team and work toward implementing them. If you run into resistance, start small.

Moving a team from dysfunctional to highly-functioning can be done with a focused and sustained effort. It's well worth the investment. Good luck and happy development!