



**Presents**  
**An IT Metrics and Productivity Journal Special Edition**

**Focus on Linda Westfall, Expert Software Metrics Practitioner**  
**A CAI State of the Practice Interview**  
**March, 2006**

**Biography of Linda Westfall**

Linda Westfall is the President of The Westfall Team. Her specialties include Software Quality Engineering, Metrics, Project & Risk Management, Requirements Engineering & Management, Peer Reviews, Testing, and Process Definition & Improvement. Linda has more than twenty years of experience in real time software engineering, quality and metrics.

Linda is the past Chair of the American Society for Quality (ASQ) Software Division. She has also served as the Software Division's Program Chair and Certification Chair, and on the ASQ National Certification Board. She was Co-Chair of the 6th Software Engineering Process Group National Meeting and past chair of the Association for Software Engineering Excellence.

Linda Westfall has an MBA from the University of Texas at Dallas and BS in Mathematics from Carnegie-Mellon University. She is an ASQ Certified Software Quality Engineer (CSQE) and an ASQ Certified Quality Auditor (CQA). Linda is also a Professional Engineer (PE) in Software Engineering in the state of Texas. Our interview with Linda Westfall and Michael Milutis, the IT Metrics and Productivity Institute Executive Director, took place in November of 2005.

◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆

**CAI: Could you tell us just a little bit about yourself, the path your career has taken, and what you're currently working on today?**

**LINDA WESTFALL:** I started out with a Mathematics degree from Carnegie Mellon University and shortly thereafter began working towards a Masters degree in Computer Science. However, there was such a demand for programmers at the time – this was in

the late 1970s - that before the end of my second semester people were already recruiting me into the workforce.

I wound up working for several small companies that were doing Department of Defense related contracts and as a result I learned a lot about good software processes right out of the gate. Soon after that, I had an opportunity to work within some real "garage shop" environments and this made it clear to me that it was the more structured kind of software engineering that I was interested in, as opposed to the ad-hoc approach.

In my next job I went right to the vice president of engineering and said, "You need somebody to keep your developers developing." This got me into doing a lot of reading and research about process. Consequently, when I left there and went to DSC Communications, I went to work in what they called "the lifecycle committee," which was a software engineering process group. DSC was about to begin a major process journey and the committee was busy establishing and defining DSC's corporate level software processes. We then moved on to put a metrics program together. I became DSC's metrics analysts and it was at this point that I really started doing metrics for a profession full-time. This was around 1988. About 6 years ago I started my own company, the Westfall Team, which does software quality engineering training and consulting. Along the course of my career, I've also gotten very involved with the American Society for Quality's Software Division. In fact, I helped chair their effort to create a Certified Software Quality Engineer certification and later became chair of their software division. Currently, my focus is on research, training, and consulting in the software quality arena.

**CAI: You write and speak a lot about software measurement. Why are good metrics so important to software project success?**

**LINDA WESTFALL:** You can't control what you don't measure. That's a quote from Tom DeMarco from years ago.

As we move from software as a craft to software as a true engineering discipline, you will find that the keystone of good engineering practices is having measurements in

place. Your measurements are the eyes and ears of your engineers, of your practitioners, and of your managers. They are your eyes and ears into what is really going on. Do you want to make decisions based upon facts as opposed to wild guesses and gut instincts? If so, you need metrics. Metrics are like an early warning system. They can tell us when things are going wrong - or right - usually long before the gut feel kicks in.

**CAI: Would you be able to quantify for us the benefits - in terms of cost savings, process completion rates and ROI - of having a solid software metrics program in place?**

**LINDA WESTFALL:** To be perfectly honest with you, I don't have any actual data on that. That's because putting metrics in place is always part of a much larger process improvement program. It's sort of a sub-component of basic process improvement. The good news, though, is that there is plenty of data on the ROI of software process improvement (SPI). If you are interested in this data, I would direct you to some of the numbers that have come out of Herb Krasner's research.

You will find a lot of CMM Level 1 organizations spending over 60% of their development dollars on the cost of quality. At CMM Level 1, most of this "cost of quality" revolves around the finding and fixing of defects. Prevention is also a cost of quality, but companies at CMM Level 1 are usually spending very little or nothing on prevention.

Think about that. Up to 60% of development dollars are being directed towards the finding and fixing of defects. This goes straight to the bottom line. Metrics are certainly part of the solution to this problem however, they don't stand alone. Ultimately, results are going to come from having mature processes in place, with a good metrics program underlying these processes.

**CAI: What are the essential components of a mature process that you cannot do without?**

**LINDA WESTFALL:** You must have ways of tracking and controlling and you must have visibility into what's going on in your processes. This is critical if you want to know what's going well for you and what's going not so well for you. Having visibility, tracking, and control will also help you target your improvement efforts to the right areas.

Understanding your processes is important, too. This means documenting your processes and disseminating that documentation throughout your organization.

**CAI: What are some of the biggest mistakes made when organizations first get started with a measurement program?**

**LINDA WESTFALL:** There are three big mistakes that people make when they first try to put metrics in place.

The first mistake is, "Let's collect data on everything, there's got to be information in there somewhere." This is like the little girl who is seriously shoveling away at the large pile of horse manure in the barn. With that much horse manure, there's got to be a pony somewhere. With that much data, there's got to be practical information somewhere. The result, though, is that people drown themselves in data.

A second mistake is when people go to a conference or they read a book or a paper and they say, "That's an interesting metric, let's do that." And then they find another article and they say, "Let's do this." They end up with a shotgun approach to metrics rather than a truly cohesive program.

A third mistake is when people do metrics for metrics sake, process for process sake. I see so many people move from having no process to having bureaucracy. That's process for process sake, metrics for metrics sake. Nobody winds up using the metrics to make a decision, but they still feel that they have to have a metrics program. The goal should never be to put a metrics program in place. The goal should always be to provide people with information so they can make better decisions.

I always recommend that people focus first on who their metrics customers are. Figure that out first, focus on what they – your metrics customers – are trying to accomplish

and what their goals are.

Keep in mind, too, that people hate to be measured. There is always an inhibition to being measured. Consequently, when I am out in the engineering and management community, I never talk about metrics. I talk about "information." I find out what "information" people need and then I put the necessary measurements in place to help provide that data.

**CAI: For those organizations that tend to be successful with their metrics program, what is it that they are doing right?**

**LINDA WESTFALL:** It's just like software development. You need to go through requirements and design before you get to coding. In metrics, you need to talk to your information customers, find out what they need to know, and then base the requirements for your metrics program on your customers' needs. So many people try to put metrics in place without talking to their customers. And then they end up with metrics that nobody uses to make decisions.

After you get the requirements phase of your metrics program complete, you must go through a "design" phase. This is what my "12 Steps" paper focuses on. The first four steps are the requirements phase. Steps 7-12 are design. Design entails going out and thinking about what the equations need to be, identifying the specific data items, finding out where and how you are going to get these data items, identifying who needs to be collecting them, determining how you are going to analyze this data once it is collected, and formulating a mechanism for getting this information to your metrics customers in an easy and understandable format. There is also a human dimension that needs to be taken into consideration. Where might people get resistant? What negative behavior might result and how can this be minimized? All of this should be part of your "design" effort.

**CAI: How would this advice of yours differ for small organizations as opposed to large organizations?**

**LINDA WESTFALL:** By their very nature, the communication channels in smaller organizations tend to be easier. Consequently, a very small organization will usually require fewer metrics because there is already a lot of visibility there. The larger the organization, the more formalized the communication channels, the more formalized the information path. Larger organizations, therefore, require much larger metrics programs.

What I always recommend for smaller organizations – because they typically don't have the resources to devote to things like metrics – is to look for key vital signs. Every time you go to the doctor your vital signs are checked: your temperature, your blood pressure, your pulse. These are your key vital signs. They tell you right away whether or not you are healthy and whether or not there is a problem. Smaller organizations should identify their key software quality vital signs.

Generally speaking, a small organization will at least want to be tracking their rework costs; specifically, the effort hours of actually doing the rework or the cost in dollars. Small organizations will also want to be tracking their schedule metrics. That would entail asking questions about process cycle times. A third vital sign area revolves around the quality and quantity of the products that are coming out of one's processes. Small organizations should pick a few key metrics out of each of these areas.

Large organizations need to do the same thing, but because they have so many projects going on, they also need to have more roll-up metrics. Large organizations need ways of pulling all of this data into more strategic, higher-level measurements that can be used to demonstrate, to upper management, the health of an organization or of a major department.

Nevertheless, whether you are operating within a large or a small organization, the fundamental processes behind the twelve steps are still the same.

**CAI: Would you expect an organization's approach to metrics implementation and process implementation to be different for maintenance based work as opposed to new development?**

**LINDA WESTFALL:** First of all, maintenance people are going to have different goals. Typically, when you're doing some kind of maintenance work, you will have a lot of very tiny projects that are focused on things like defects and rework. You won't be dealing with project management type metrics like value or schedule actuals vs. budget. You will also be much more focused on post-release as opposed to pre-release defects.

Another thing about maintenance is that the metrics are going to be fed back to the development community. The maintenance people are going to look at the root causes of the defects they're fixing and communicate this back to the developers.

**CAI: In light of the fact that process and metrics are so interdependent, what must organizations first have in place on the process side before they can expect to make any progress with their metrics efforts?**

**LINDA WESTFALL:** If you don't have consistent repeatable processes in place, the amount of variation you're going to see in your metrics will be so large that you may actually need to start measuring the variation. This is one of the biggest frustrations I personally encounter with companies. Your processes really do have to be repeatable. Otherwise, you'll be measuring apples and oranges and calling it fruit salad.

Keep in mind that your metrics themselves will mature as your processes mature. One of the things I always recommend is that if you have processes that are very immature, you should look at metrics that revolve around improving these processes. For example, if you've just started making an effort to implement peer reviews, you might want to ask yourself whether or not you are even holding these reviews, what percentage of your work projects are getting peer reviewed, or how much you are spending on peer reviews in dollars. As your processes become more mature, you will want to start looking for metrics that are going to help you optimize your processes. To this end, and to continue with the peer review example, you might want to ask questions about your optimum inspection rate, questions about how many pages or lines of code per hour you are from perfect, even questions about how many people you should put in the room. We did some studies like this at DSC Communication, and

found out that if we were trying to emphasize efficiency, the ideal was to put four people in the room. But if we needed to identify the maximum amount of defects – usually because we were dealing with a critical or high-risk piece of code - then were better off putting six people in the room. It wasn't as efficient but it was more effective. That's because the fifth person and the sixth person each found more defects than the other four didn't. When we added the seventh person, they didn't find anything that the other six hadn't also found.

When you get even more mature you start putting metrics in place that are looking for process improvement opportunities. Where are my road blocks in this process? Where am I wasting time? Where am I doing unnecessary rework? What defects are escaping from this process?

My point is that the kinds of metrics you're going to be looking at will change over time as your processes start to mature. They will mature right alongside your processes.

Questions? Suggestions? Comments? Please contact the IT Metrics and Productivity