



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

**Focus on Manfred Bundschuh
President of the German Software Metrics Association
Co-author of *Best Practices in Software Measurement*
A CAI State of the Practice Interview
August, 2006**

Biography of Manfred Bundschuh

Manfred Bundschuh is the co-author — along with Christof Ebert, Andreas Schmietendorf, and Reiner Dumke — of *Best Practices in Software Measurement*. For the past 20 years he has been the quality manager of AXA Service AG in Cologne, Germany. In 1983 he was appointed professor for software engineering and project management at the University of Applied Sciences in Cologne. Mr. Bundschuh is also president of the Deutschsprachige Anwendergruppe für Softwaremetrik und Aufwandschätzung (DASMA). He lectures for various organizations and has published more than 40 papers (some in books) and 9 books (3 as co-publisher). Our interview between Manfred Bundschuh and Michael Milutis, Executive Director of the IT Metrics and Productivity Institute, was conducted in April of 2006.

◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆

CAI: Could you tell us a little about yourself, your background and what you are working on today?

MANFRED BUNDSCHUH: For the past 23 years, I have been a Professor for Software Engineering and Project Management at the University of Applied Sciences in Cologne, Germany. I have been a board member of the German Software Metric Association (DASMA) for the past 10 years and its president for the past 6 years. In the 1980s and 1990s I was Chairman of the Cologne Chamber of Commerce (IHK).

Since 1983, I have been the IT quality manager at AXA, the second largest insurance

company in Germany and one of the five largest in the world. I am responsible for the education and training of the IT staff and junior programmers. After my introduction to function points in 1994, I initiated function point counting and measurement at AXA, which was developed into an integrated multi-project management system.

CAI: The Standish Group reported in 2000 that approximately 70% of all software projects come in either over budget, over schedule or not at all. What do you make of these statistics?

MANFRED BUNDSCHUH: The 70% failure rate is only a mean value. The best performers have their projects completed on time, on budget and on specification. However, the statistic certainly implies an overall lack of communication, as well as a lack of clarity around goals and requirements. That's the first general problem.

The second problem is over-optimistic estimates or, in some cases, lack of any estimates at all. A professional project management team must make their measurement estimates at the very earliest stages of planning. Nevertheless, project architects and project leaders will often claim that they have no time to calculate estimates. This is simply unprofessional. Estimates of size metrics and effort metrics are very important for getting started and gaining insight. Metrics also have a value for benchmarking.

To answer the second part of your question, what these statistics tell us is that the art of modern software engineering is typically practiced at the expense of the client. I like to call this malpractice by management.

CAI: How does your book - *Best Practices in Software Measurement* - address some of these issues? What did you set out to achieve when you initially wrote it?

MANFRED BUNDSCHUH: Measurement can be a double-edged sword. It can save money and time, or it can cause damage to an organization. Organizations that border on a culture of malpractice will use numbers and statistics to say anything they want.

On the other hand, the best performers tend to utilize their software metrics data properly.

The mission of this book was to document many of these actual cases. I wanted to provide examples of success stories in which software metrics had been properly exploited. From an educational perspective, I believe it is very important to present examples in the form of case studies. Case studies make learning very easy. Consequently, we have made use of many case studies in our book from some of the largest organizations in Germany.

CAI: You make use of this term "malpractice" quite a lot. Do you mean this literally or figuratively?

MANFRED BUNDSCHUH: I specifically want people to hear the word "malpractice." It should shake people up a bit. The fact is, organizations have no idea how inefficient and unproductive they are. They do not take the time or effort to learn about best practices. As a result, they continue to operate in the dark; uninformed of all the various methods that could be having a positive impact on their business.

CAI: Could you qualify for our readers the "positive impact" of having a reliable software metrics program?

MANFRED BUNDSCHUH: The primary positive impact of a software metrics program is manageable IT; that is to say, a reliable software metrics program will give you *control*. You will have controls on your processes, controls on your productivity and, ultimately, controls on your quality. A reliable software metrics program will also give you the ability to benchmark.

CAI: Once an organization has been successful in integrating a reliable metrics programs into their software operations, how do they go about quantifying the "positive impact" in terms of ROI?

MANFRED BUNDSCHUH: ROI is an all important concept in software process improvement. Actual efforts have to be examined. At the same time, returns must be projected for scenarios in which the same hypothetical capital had been invested elsewhere.

In chapter 9.6 of my book, starting from page 150, we actually deal with this very issue. We provide some guidelines for ROI calculations and we even document actual experiences.

CAI: What arguments do organizations tend to make against the use of software metrics? What would be your best response to those arguments?

MANFRED BUNDSCHUH: Many organizations develop a very unhealthy attitude about software metrics. One of our major challenges is helping management understand the true risk/reward factors.

One argument against using metrics, for example, is that they are too time consuming. In reality, however, we have found that estimations for small and medium sized projects normally range between five and ten hours. Larger projects may take a few days, but this is negligible compared to the amount of time spent on the entire project as a whole.

A second argument against using metrics is that measurement itself is too expensive. Our experience, however, is that the effort and cost of a metrics program will be negligible when compared to the effort and cost expended on the entire project as a whole. Moreover, the added value more than compensates for the purported additional effort. In fact, it is going to be much more expensive in the long run for an organization to neglect proper measurement and estimation.

An argument that is leveled against function points, specifically, is that they were designed by theoreticians - for use by practitioners. This argument, however, is neither accurate nor legitimate. Function points were developed in response to very practical needs, not some sort of theoretical philosophy.

CAI: What exactly are function points? Could you give us a definition?

MANFRED BUNDSCHUH: Function points are a method for measuring the relative size and complexity of a software system. Function points are generally measured by counting the various functional components of the software. An example of this might be to count the number of inputs or the number of interfaces.

Function points are the basis for many metrics. Some examples might include quality metrics (such as error per function point) or team size metrics (such as employees per function point). Other function point metrics include productivity metrics, maintenance metrics, effort metrics and delivery metrics.

Function points have many benefits, not least of which is the role they play in estimation. A project leader once approached me and said, "I have to estimate a project which starts next year. I don't know anything! Can you help me?" I began by asking a series of function point based questions. "How many interfaces do you have? How many inputs and outputs do you expect?" Combining their responses with the data available on these functional areas allowed me to provide them with some fairly accurate estimates.

CAI: Why do so many measurement programs fail? Could you summarize for us some of the most common pitfalls and how to avoid them?

MANFRED BUNDSCHUH: Many metric projects fail due to simple lack of planning. From the very earliest stages of projects there is inadequate planning for measurement, inadequate planning for estimation, and little or no goal orientation. And even in cases where planning is done well, it is often carried out half-heartedly.

Another common pitfall is the improper utilization of data after it is collected. Tom DeMarco has done some studies on this subject and his conclusions demonstrate that many organizations are actually gathering metrics simply for the sake of gathering metrics. While this pointless exercise persists, other organizations are figuring out how to gain tenfold increases in productivity and quality simply by targeting their metrics to real business goals.

Yet another mistake is when estimation is performed by only one or two people. Estimation should always be done in teams. I have found that using project teams for estimation helps workers learn a great deal more about their project. They gain a better grasp of project risks during this process and, needless to say, their estimates also tend to be more accurate.

A fundamental problem at the heart of this issue is that many people simply do not know how to estimate properly. As a result, patently ridiculous estimates are often accepted by management without question. One way to avoid this trap is to only accept estimates in intervals and ranges. Another more obvious remedy is to send your employees to estimation conferences and to provide them with up to date literature.

CAI: What questions should organizations be asking themselves when trying to identify an appropriate metrics strategy?

MANFRED BUNDSCHUH: The best advice I can give is to use the Goal-Question-Metric Approach. This was originally proposed by Victor Basili at the University of Maryland. It is widely used today and very practicable.

A goal, for example, of executive management might be to control costs. Their questions would be economic in nature. Consequently, they would have an interest in effort and quality metrics.

A software developer, on the other hand, might have goals that relate to conditions or environment. In light of this, a developer's questions would be of a more technical nature and would be more suited to size and complexity metrics.

End users, yet another constituency, might have usability goals. Their questions (and the metrics that tie back to these questions) would directly relate to functionality.

CAI: It sounds like there are a lot of different metrics, and a lot of different consumers of metrics. How do organizations best choose the most appropriate measures?

MANFRED BUNDSCHUH: There are, indeed, many different metrics – some might say too many. One book by Horst Zuse - *A Framework of Software Measurement* – cites over 1000 different metrics.

The best metrics for an organization are the ones that are both easy to use and easy to measure. It is preferable to find metrics that can be measured automatically. These will often be the most standard metrics. Standard metrics have an additional benefit in that they can be used to benchmark and compare against other organizations.

Another recommendation would be to start with only a handful of measures. Size, effort, quality, cost, duration and team size would be an excellent place to start.

CAI: What questions should organizations ask themselves after they have successfully identified and selected their metrics?

MANFRED BUNDSCHUH: Do the metrics we are selecting support the questions that we are trying to answer? Are the selected metrics adequate for our business needs? Are the selected metrics producing clear measures? Are the metrics providing reliable data? What can we learn from the results? Can we use our results for benchmarking purposes? How can we improve the process of measurement?

CAI: What role, if any, do tools play in software measurement?

MANFRED BUNDSCHUH: In my own organization I created a special estimation sheet for software maintenance tasks using Microsoft Excel spreadsheets. Originally, I thought it would be highly inefficient to count function points for small maintenance tasks. Fortunately, however, one of the senior department leaders introduced me to a tool called Function Point Workbench. He showed me how easy it would be to count the function points using this tool.

For example, if I have to order a screen in an application, I can access Function Point Workbench with a few mouse clicks. I can view the screen in terms of function points, order the function points and then verify the number of function points that are

involved in ordering the screen. I can additionally determine whether or not function points have been added, deleted or edited for each task. This procedure can be repeated for each maintenance task rather easily.

My point in telling this story is that there are plenty of simple tools that can make a difference without costing a lot in terms of effort, implementation, or expense.

CAI: Aside from your own book, could you recommend other additional reading material for people that are interested in this subject?

MANFRED BUNDSCHUH: I would recommend *IT Measurement: Practical Advice from the Experts*, a book published by International Function Point Users Group. It provides a comprehensive examination of estimation and measurement. I would also recommend *Metrics and Models in Software Quality Engineering* by Dr. Stephen Kan of IBM. For those interested in this subject area, Dr. Kan's book is one of the most thorough texts available.

Questions? Suggestions? Comments? Please contact the IT Metrics and Productivity Journal Editor at michael_milutis@compaid.com