



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

**Focus on Dr. Barbara Kitchenham,
Professor of Quantitative Software Engineering, Keele University
A CAI State of the Practice Interview
September, 2006**

Biography of Dr. Barbara Kitchenham

Dr. Barbara Kitchenham is Professor of Quantitative Software Engineering at Keele University in the UK; she is also a Senior Principal Research at National ICT Australia. She has worked in software engineering for nearly 30 years both in industry and academia. Her main research interest is software metrics and its application to project management, quality control, risk management and evaluation of software technologies. She is particularly interested in the limitations of technology and the practical problems associated with applying measurement technologies and experimental methods to software engineering. She is a Chartered Mathematician and Fellow of the Institute of Mathematics and Its Applications. She is also a Fellow of the Royal Statistical Society and a visiting professor at both the University of Durham and the University of Ulster.

Our interview between Barbara Kitchenham and Michael Milutis, Executive Director of the IT Metrics and Productivity Institute, was conducted in June of 2006.



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

BARBARA KITCHENHAM: My university training is as a mathematician/statistician. My first job after obtaining my PhD was actually as a statistician in a research institute. However, when my husband moved for his job in computer science, I followed him into the computer industry and became a programmer-designer at ICL (later SPC and then Fujitsu).

At the time, Professor Manny Lehman was publishing his research on system evolution.

He wrote reports for my workgroup at ICL that contained terms such as “random error.” Because I was the only person in the group with a background in statistics, they were given to me to read. These reports laid the groundwork for my later interest in measurement and estimation. By the late 1970s and early 1980s the industry had developed more interest in metrics and in the improvement of estimation practices and I wound up becoming the point person for that at ICL, too. It was also about this time that I began working on government research projects.

After working at ICL I spent a couple of years at the Center for Software Reliability at City University, and I worked for eight years with the National Computing Center. I now work part time for Keele University in the UK and part time for National ICT in Australia.

CAI: The Standish Group reported in 2000 that over 70% of software projects were coming in over time, over budget, or not at all. Do you have any thoughts on what we can do to improve project estimates in general, in light of these bad numbers?

BARBARA KITCHENHAM: Without knowing the methodology of the Standish Group study, it is difficult to say if these numbers are accurate or not. First of all, we don't have a very good definition of what constitutes a successful project. Second, budget and time scales are easy to manipulate and sometimes political in nature; they may not be the best criteria for deciding whether or not a project is successful. Finally, it is difficult to make statements about the industry as a whole unless we can be sure that we have a statistically random sample of companies throughout the industry. You have to keep in mind that the companies that answer these kinds of surveys are often the same ones that have problems.

Looking at those companies that do have problems with estimates, it is hard to say why their projects are under or over budget. Is it because the project didn't go well or because the estimates in the budget were wrong in the first place? If you don't know the fundamental problem, it's going to be difficult to understand what these numbers mean.

My educated guess is that the estimates are not necessarily wrong. People may make good or bad estimates, The problem is that those estimates are then converted into contractual bids which can be manipulated by senior managers. And it's perfectly respectable for a senior manager to submit a bid lower than the estimate given by the development staff. People don't necessarily run their budgets from their estimates. They run their budgets from their contracts and these contracts may or may not be a reflection of the original development team's estimates.

Keep in mind that the company may be happy to overrun by 10% if their profit rate is 30%. I've certainly spoken to senior people in the industry who say, "We always give people a tight schedule and budget because it makes them work harder." When you fail under these kinds of circumstances, is it really a failure?

CAI: Estimation always seems to come back to a discussion of measurement. Can you compare and contrast some of the existing measurement methods out there and in use today?

BARBARA KITCHENHAM: Function points are a pretty good metric for estimating, particularly if you're concerned about estimating what your own company is doing.

I don't think function points should be used for cross-company benchmarking, though. That's because the measures are not objective and independent, the way meters or kilograms are. They are dependent on the nature of the products that your company produces, the way your people work, and the cultural environment in which you build your software.

However, looking at lines of code is a good way of checking how well you're doing and assessing your capabilities. It is not as good for estimating - because it's hard to know from the start of a project what the code is going to be like - but it's good for assessing your capabilities.

Object-oriented metrics are best in the design phase but I don't think they make very sensible measures for cost estimation. Like lines of code, they are difficult to determine at the outset of a project when you don't know yet how many classes and

objects you will have. However, if a company does applications where other people create requirements, and they're bidding for the design and implementation phases, then perhaps in these cases object-oriented metrics provide a suitable level of granularity for the kinds of estimates that are needed.

CAI: Capers Jones once wrote that, "Measurements, metrics, and statistical analysis of data are the basic tools of science and engineering. Unfortunately, the software industry has existed for more than 50 years with metrics that have never been formally validated and with statistical techniques that are at best questionable." How do you feel about that statement?

BARBARA KITCHENHAM: I agree with Capers completely. Most of my time is spent working with people to formulate better analyses and improve sub-optimal metrics.

An underlying problem is that academic software engineering departments do not allow much time in their course programs for statistics or experimental methods, and yet we still want our software engineers to come out of university on top of all the techniques associated with math and science. There is also a lack of interest in the research community in multi-disciplinary work in general.

I agree that our metrics have not been formally validated. What we have are measures that work reasonably well in particular contexts. Because we want to think we're real engineers, we want our measures to be context independent, but I don't think that's possible, or even necessarily important.

CAI: You mentioned some of the challenges with what you called cross-company benchmarking. How do we address that problem? Is there a way to benchmark effectively across companies?

BARBARA KITCHENHAM: If we could do formal stratified randomized sampling, then maybe we could make cross-company comparisons. At the moment, however, the benchmarking data available to us is not a representative sample of the industry as a whole. What we have is data from organizations that choose to volunteer their data in

order to get analyses done. Consultancy groups offering benchmarking data are likely to have a disproportionate number of companies with problems represented in their data, because in general the people who consult these groups are the people who have problems.

At one time we used to have advertisements that said, "5 out of 10 cats prefer Kitty Cat." Advertising standards won't let people say that now. They can only say, "5 out of 10 cats that were tested prefer Kitty Cat." It seems to me that our benchmarking databases are still self-selecting in the same manner as the old Kitty Cat ads.

However, comparing yourself across companies in your industry is one of those management obsessions that in my opinion can be counter-productive. Trying to always compare yourself to your competitors is not a wise use of energy. It takes effort away from what you really want to do, which is to make sure your company is estimating well, getting bids, and making money.

CAI: There are many different functional size methodologies out there today, the most well known of which are IFPUG function points, Mark II function points, and COSMIC function points. What are the distinguishing characteristics of each of these variations? What are the relative advantages and disadvantages?

BARBARA KITCHENHAM: I am not an expert on the COSMIC methodology. However, I took a considerable interest some years ago in the difference between the standard Albrecht (IFPUG) and Mark II function point methods. Speaking as a mathematician, I think the Albrecht method uses numbers incorrectly. It's not good practice to convert things into an ordinal scale and then start adding and multiplying them together again. On those grounds I prefer the Mark II methodology, which uses simple counts that require a weighted addition. Although we don't have any theoretical models to justify the weights (nor do we have any way of empirically determining universal weights, because we cannot effectively sample the population) I think you can probably find the correct weights for your own organization.

I don't like complexity adjustments, which add confusion between size and complexity measures. If you run function points through a backfiring and then bring them into COCOMO, you've applied the adjustment factors a couple of times over. Consequently, it seems to me that you can't really understand what your estimate is going to be about.

Ultimately, I haven't seen a lot of evidence that complex estimations are any better than simple ones for moderate to small sized applications. For large applications, it's hard to say. That's because there is an extremely limited amount of data to study. There are only a few projects or products that are the size of Microsoft Word.

CAI: Capers Jones writes that, "Function points are not perfect, but they have provided more useful data and results than most alternative methods." Do you agree with this statement?

BARBARA KITCHENHAM: I agree with that statement. I also think lines of code can be surprisingly effective as long as you treat them with caution. What's funny is that we know we are supposed to be cautious about lines of codes, but we seem to forget to be cautious about function points. I don't know why we forget to be cautious. Perhaps people think that because function points are a bit more complicated they are bound to be better. Or perhaps we have the strange idea that programmers can't fudge function points in the way that they can fudge lines of code. Most programmers are really, by nature, quite clever. If they want to produce an application with lots of function points, they will probably find a way to do it.

CAI: What in your opinion does the IT industry need to do over the next 5-10 years to transform their software measurement practices into a discipline as rigorous as that found in other engineering fields?

BARBARA KITCHENHAM: I'm not sure we would want software engineering metrics to be a science like electrical or mechanical engineering metrics. That might be a bit of wasted effort. We should focus our efforts on understanding the measures we have,

understanding their limitations, and using them constructively within those limitations.

I don't think we should have international standards for counting function points. That's because we don't have for function points the equivalent of the platinum bar that tells us, for instance, what a meter is. For the same reason, I don't think function point metrics should be quoted in courtrooms. However, I do think we can have agreed conventions for counting function points.

I think you can do cross-company comparisons if you really must. However, I would be a bit cautious about the results. I would rather see people using the measures that they can get, with caution, in their own environment in order to help themselves get better.

I certainly would never recommend that people estimate solely with an estimating model. They should use at least two forms of independent estimation methods. A model may be one of those, but they should also use expert opinion – and afterwards compare the results. Looking at studies from the past 20 years, it's quite clear that when models are compared with expert opinion, the numbers don't always win. In short, I wouldn't trust my billions of dollars on COCOMO alone. I'd want to use COCOMO or another estimating model, and then I'd also want an expert opinion estimate. The estimation process should also be separated from the business requirements process so that if requirements are adjusted for business necessities or for client requests, the estimation process will stand independently.

In practical terms, use reasonable measures and use them with caution. I would also concentrate on making sure I had an estimation process, not a model. And I would focus on my own company and what it was doing and would be hesitant about spending money and energy on cross-company benchmarking. Clearly you can't avoid that entirely in a competitive world, but it's important to know what you are doing well or poorly in your own company before you start looking at what other people are doing.

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