



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

**Focus on Barry Boehm, Software Estimation Pioneer**  
**A CAI State of the Practice Interview**  
**August, 2006**

**Biography of Barry Boehm**

Barry Boehm received his B.A. degree from Harvard in 1957, and his M.S. and Ph.D. degrees from UCLA in 1961 and 1964, all in Mathematics. Between 1989 and 1992, he served within the U.S. Department of Defense (DoD) as Director of the DARPA Information Science and Technology Office, and as Director of the DDR&E Software and Computer Technology Office. His current research interests include software process modeling, software requirements engineering, software architectures, software metrics and cost models, software engineering environments, and knowledge-based software engineering. His contributions to the field include the Constructive Cost Model (COCOMO), the Spiral Model of the software process, the Theory W (win-win) approach to software management and requirements determination and two advanced software engineering environments: the TRW Software Productivity System and Quantum Leap Environment. Our interview between Barry Boehm and Michael Milutis, Executive Director of the IT Metrics and Productivity Institute, took place in April of 2006.

◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆

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

**BOEHM:** I began my career in 1955 as an analyst at General Dynamics, doing computer programming for rocket related problems. In 1959 I went to the Rand Corporation, where I remained until 1973. I spent my first two years at Rand developing software and my second two years doing system analysis. Ironically, my work in systems analysis identified software as a critical problem for the company. Consequently, I went back to developing software again. In 1973 I moved on to TRW,

where I remained until 1989.

While I was chief scientist at TRW the chair manager there maintained that we had no rational way of estimating software costs – and that this was our biggest problem. He asked me if I could do something about it. That led to a TRW project that eventually became the basis for the COCOMO cost-estimating model. I left TRW in 1989 in order to run the DARPA Information Science & Technology office for the government. Three years later I came to USC.

My work today primarily revolves around the integration of cost analysis and benefit analysis into something we call “value-based software and system engineering.” In all of your projects, you should always consider the relative value of any proposed task. Most software engineering projects view every requirement as equally important – every use case, every object test case, every defect. And that ends up wasting a lot of time and resources. Our work with value based engineering is an attempt to develop guidelines to help improve this process.

**CAI: Has any of this value-based engineering taken root yet in the world of corporate IT, or is it still in the early research stages?**

**BOEHM:** It’s being practiced in a lot of places. The financial institutions have been using this approach all along. But most other players need to catch up, especially in the area of software methodology. Our goal at USC is to integrate what business and financial people know about valuation, risk analysis and risk management and to apply this knowledge to software development.

**CAI: You are most well known for your work with software estimation. Generally speaking, what are the primary factors that influence software cost and how can we use this knowledge to improve software estimation?**

**BOEHM:** The two primary factors that influence software cost are the size of the project and the people involved in the project. In regard to size, you will want to prioritize features based on value and then address these first. You can then re-

evaluate whether you really need the other remaining features. This will prevent you from tackling all the specifications at once, building everything, only to learn later that you are only using 25% of what you developed.

The people factor revolves around capability, experience, collaboration, and retention. You'll usually have a productivity factor difference of over 10 if you're doing the same job with qualified as opposed to unqualified people.

**CAI: In the 1980's you wrote a book, *Software Engineering Economics*, that is considered by many to be the best book on software estimation yet written. Could you tell us what motivated you to write this book and what you set out to achieve with it?**

**BOEHM:** When I was at TRW in 1978, I recognized the need for a textbook on software engineering. So I took a sabbatical from TRW and went to USC to start writing. By the end of the semester, I had completed a few introductory chapters and a couple of in depth chapters on economics and cost estimation. But I still had about 20 large chapters that were unfinished. What I realized, at this point, was that my original scope for the book was just too big. Consequently, I narrowed the focus to software economics and cost estimation alone. Several years later, the book was published.

**CAI: In *Software Engineering Economics* you write about decision making techniques that come into play with information systems development. What techniques and tools are available today for making good decisions?**

**BOEHM:** The key thing in decision making is to bring your stakeholders together and to have them identify and prioritize the features and attributes that are most important to a project. We have a collaborative support tool that we use for this called Easy Win-Win. It's now commercially supported by a company called Group Systems. Easy Win-Win lets all the stakeholders rapidly input their win conditions and share them among the entire group. Once all the stakeholders understand each other's win conditions, issues can be identified, negotiated and resolved, making it a "win-win" for all

stakeholders.

There is a piece in this software that asks users and clients to prioritize the business importance of the features they are requesting on a scale of 1 to 10. The developers are then asked to rank the ease of development for the requested features. Consequently, if you get a feature that is identified as a 10-10, meaning something that is both important and easy to implement, it's clearly the type of attribute you would want to develop first.

**CAI: In *Software Engineering Economics* you also write that good software engineering must accommodate human concerns. Could you elaborate on this?**

**BOEHM:** There are two parts to this concept. The first part revolves around understanding the people who are actually using your product. From a programming perspective, we have to unlearn the Golden Rule, which says "do unto others as you would have others do unto you." As a programmer, the literal interpretation of that rule is to give others a programmer-friendly GUI. But that's not going to be very useful to a nurse or a librarian. Getting software people to understand this idea is difficult. We emphasize prototyping and a lot of client mutual learning to avoid this problem.

The second part of this concept revolves around understanding the people who are doing the development work. You have to think about their needs, too. Tom DeMarco and Tim Lister's *PeopleWare* provides loads of example on how companies often make short-sighted, bean-counting decisions that adversely impact development work – for instance, putting people in undersized cubicles with lots of noise and interruptions and no opportunity to get continuity of thought. In the end, your people will get frustrated and leave. The SEI's People Capability Maturity Model can help you avoid some of these problems.

**CAI: Estimation, process and measurement is fairly tightly interconnected. In light of this, what do organizations need to have in place on the process and metrics side before they can expect to see any return on their estimation**

**efforts?**

**BOEHM:** The most important thing is good data definition. Before organizations rush to empty their filing cabinets or to throw information into a calibration database, they need to make sure that the data they are inputting is consistently defined.

**CAI: Can developers profit by employing use cases as a functional definition for their estimation efforts?**

**BOEHM:** We've done some experiments with use cases and always end up getting about a 0.4 to 0.5 correlation between use cases (weighted by complexity or number of elements in the sequence diagram). In reality, you want to see something in the range of 0.8. We find that the use cases frequently overlap. Also, they aren't as well defined as function points, in terms of counting rules. If you go to a function point counting course, you'll usually come out within 15% of somebody else who is counting function points.

**CAI: The Standish Group reported in the year 2000 that over 70% of software projects were coming in over time, over budget, or not at all. What is the current state of project estimation in 2006? What in your opinion needs to happen over the next 10 years for us to make significant progress with these numbers?**

**BOEHM:** The biggest problem we have today is that there are still many developers who don't use cost models. What that means is that these developers have no experience-based, facts-based method for countering people with unrealistic expectations. Most developers just want to avoid conflict. Moreover, they tend to think optimistically, promising to do things that they really shouldn't promise to do – and certainly wouldn't promise if they really understood what they were doing.

When you're dealing with relatively small and independent projects, things like agile methods can certainly help avoid these problems. With agile methods, you can say that you are going to build something in a month. If your stakeholders don't think it does

what they want, the next month you can provide them with another version. What this means is that, if you miss the target, you've only lost an investment of a month. You'll get much more rapid feedback from your stakeholders and you can still get the project back on track without too much difficulty if there is a problem.

**CAI: With so many cost models available, how can organizations determine what models they should be using? Are there guidelines that you can recommend?**

**BOEHM:** Most organizations have a primary clientele, so most of the time you will be trying to match a solution with an industry. For example, Capers Jones' models are primarily targeted toward business data processing. Models like SEERS, on the other hand, tend to be directed toward aerospace applications. COCOMO is skewed toward aerospace, telecommunications and infrastructure.

**CAI: You first published the COCOMO cost model in 1981. Then, in the 1990s, it was reinvented and became COCOMO 2. Could you talk about what makes the COCOMO cost estimation model unique among other models?**

**BOEHM:** COCOMO's primary distinction is that it is completely open and public. In order to remain in business, commercial proprietary models can't give away their tables, equations and parameter values. But with COCOMO, we put all of this information in the book and on the CD. Anybody who wants to leverage this for commercial purposes is welcome to do so. In fact, several companies have done just that. Some of the estimation proprietary models like Price-S or True-S have a COCOMO component that you can exercise as a part of their product.

**CAI: Could you talk a bit more about the current state of estimation, specifically in terms of modern, object oriented development environments?**

**BOEHM:** When you are dealing with a higher level set of programming constructs, it becomes much more difficult to define what size really means. For instance, if you are doing Java server pages, GUI builders, or similar work, counting instructions are not really going to be relevant.

In light of this, I think the best option we have right now is function points. You can use some kind of backfire ratio from the function points and relate it to whatever language level you're operating in. After that, you can factor in variables such as re-use and requirements follow-ability, which will help you to come up with the source instructions for what you are developing.

**CAI: For organizations that are experiencing chaos in terms of their process and measuring capabilities, how would you advise them to get started?**

**BOEHM:** I think the best single reference for getting started is a book that was produced by Cheryl Jones called *Practical Systems Measurement*. It's fundamental in a very pragmatic way. It teaches the basics for getting started, the main quantities that you want to be able to define with candidate definitions, and different measurement techniques as well. Most of the cost models have assumptions built into them about what they do and don't count, about where they start and stop. If your process more or less fits an existing model, you should start with the model and then make use of the book for guidance.

Questions? Suggestions? Comments? Please contact the IT Metrics and Productivity Journal Editor at [michael\\_milutis@compaid.com](mailto:michael_milutis@compaid.com)