



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

**Focus on Michael Mah
A CAI State of the Practice Interview
December, 2006**

Biography of Michael Mah

Michael Mah is a Senior Consultant with Cutter Consortium's Business Technology Trends & Impacts, Measurement and Benchmarking, Agile Software Development & Project Management, and Sourcing & Vendor Relationships Practices. He is also owner/partner at QSM Associates Inc. Mr. Mah is a recognized expert on practical applications of software metrics, project estimation/control, and IT productivity benchmarking. Over the past 10 years, he has published numerous articles on these and other management topics. His recent work merges concepts in software measurement and benchmarking with negotiation and dispute resolution techniques for IT outsourcing and relationship management. Mr. Mah's particular interest is in people dynamics, such as the complex interactions between people, groups, divisions, and partnered companies working on the technology revolution at "Internet speed." He is also focused on the latest research and theory on negotiation, including the use of game theory, role playing, and training to increase corporate and personal effectiveness. Our interview between Michael Milutis, Executive Director of the IT Metrics and Productivity Institute, and Michael Mah took place in August of 2006.

◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆

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

MICHAEL MAH: My educational training was in physics and engineering. Specifically, I studied electromagnetic field theory and electrical engineering at Tufts University. Shortly after finishing my undergraduate engineering work, I was hired by the Trident Nuclear Submarine Program.

The work I did for Trident steered me into the field of software. At Trident, I got involved with the navigational systems of large nuclear submarines, part of the powerful U.S. strategic deterrence arsenal at the time. At the heart of it all was very complicated forecasting software that enabled the submarines to navigate correctly. I was the managing group leader charged with integration testing for much of this software.

A great deal was at stake making sure that these ships flew correctly. Consequently, I developed a fascination with the art of forecasting and with the need for prediction accuracy in large-scale, highly complex projects.

It was at about this time that I met Larry Putnam after following some of his research in several IEEE journals. Working together, we wound up being able to jointly predict how the submarine project would unfold – and we did it with better than 90% accuracy. That’s how I got “religious” about the importance of being able to anticipate the future.

After spending a few more years on the submarine program, I was offered a job at QSM - the company Mr. Putnam founded in 1978. I am currently one of their managing partners. We have several branches worldwide including offices in Europe, Asia and North America. Together, we garner an incredible amount of intelligence about the nature of technology and how software is utilized worldwide.

I should also mention that, aside from my background in engineering and physics, I studied conflict theory and mediation at the Program on Negotiation, which is a consortium of Tufts University, Harvard University and MIT that focuses on the theory and practice of conflict resolution. All of this started after a bad experience from an outsourcing relationship that I was personally involved in. The disagreements revolved around completion dates, productivity targets and cost reduction. As a result of this experience, and after talking with a colleague in the software business who had become a full time arbitrator, I decided to formally pursue a path in mediation. Since then I have been able to help many companies negotiate outsourcing agreements, in particular when estimation and productivity are at the forefront of contractual discussions.

CAI: The Standish Group came out with a study several years ago that showed 70% of all software projects coming in over budget, over schedule, or not at all. What do you attribute this problem to?

MICHAEL MAH: I take issue with the Standish report.

First of all, their definition of failure implicitly assumes that the original deadlines and budgets were reasonable to begin with. In all sincerity, many of the projects that I have been involved with have had deadlines and budgets that were somewhat arbitrary.

When I speak to hundreds of people at conferences and ask, "How many of you, when asked to work on a critical project, have had the completion date imposed on you from the very beginning?," I can tell you that approximately 95% of the hands go up. However, when I ask these people if they had a 50-50 chance of completing these same projects within the given timeframe, about one-quarter of the hands go up. That tells you something.

Another problem with the Standish Report is their interpretation of the data. For example, I don't think it would be at all unreasonable to suggest that if a project is working against an impossible deadline and is being executed poorly, that canceling the project would be a wise decision. Likewise, the missing of an arbitrary deadline, in order to deliver more reliable functionality in a more reasonable timeframe is something that, in my opinion, should be viewed as a criterion for success in many cases – not failure. In fact, many managers maintain that they make these types of conscious decisions all the time.

Larry Putnam has eloquently explained that people act as though time is something they can control, influence and manipulate. Reality, however, clearly shows that time is a very difficult variable to control. When a project manager starts banging the table and laying out a timetable, perhaps he is simply trying to influence the direction of the project in a powerful way. However, such deadlines do not – and should not – constitute the benchmark for a project's success or failure.

CAI: But if part of the problem here is that project completion dates are being chosen arbitrarily, couldn't we reasonably conclude that estimation techniques, in general, are not being used properly and that managers should be looking towards improving them? Wouldn't that be a reasonable interpretation of the Standish Report's data?

MICHAEL MAH: Figuring out the date by which a certain amount of functionality should be delivered is something that might involve estimation. However, setting a target date can also be about stimulation. Many managers will use a deadline to motivate their development team to perform. This approach has nothing to do with the forecasting of reasonable estimates.

CAI: What do people really mean when they talk about software process improvement?

MICHAEL MAH: I have my own unique perspective on this. Software process improvement, in my opinion, is really about optimizing the invention of software in a design intensive environment. And this improvement tends to be more about knowledge work and research than about the kind of process improvement you do when you are automating something that you already know how to build. In short, process improvement in knowledge work is much different than process improvement in factory work.

Consequently, software process improvement succeeds most when the power of team collaboration is realized. So whether it is in design reviews, code reviews, test coverage, QA management or configuration management, software process improvement will always be about improving the way people work together. When people understand this principle they will succeed. When people fail to grasp this principle, their attempts at process improvement will most likely fail.

CAI: Which particular software process improvement methodologies can play a role, in your opinion, in addressing some of the productivity problems that

have historically plagued our industry?

MICHAEL MAH: It seems today that the entire software engineering industry is falling into one of two religious sects. Either you belong to the "Church of CMM" (Capability Maturity Model), which is located at Carnegie Mellon University, or you are a member of the "Church of Agile," which boasts a considerable number of today's leading software thinkers. People are simply being asked to choose.

Methodologies aside, though, I believe – as I stated in your previous question - that success or failure in software process improvement will begin only once people start thinking about how they work together. If a project team is struggling to meet impossible deadlines, they will simply not have time to think about improving their teamwork. Process improvement, in these cases, will become little more than lip service. After all, how can people ever be expected to analyze and improve their teamwork when they are scrambling to get code out the door?

CAI: How does software measurement play a role in software process improvement? Why is so much importance placed on this?

MICHAEL MAH: I will employ a medical metaphor here. If you came from a family with a history of heart disease, how could you ever be expected to improve the process of exercising, eating, sleeping and working - or to determine how each of these were impacting your cholesterol level – unless past records were kept?

Measurements are a diagnostic instrument. In the field of medicine, they are able to give accurate readings of what is going on in the body. In the case of IT, the "body" is either the software, the project, the team, or the company.

Measurement, however, requires time, effort and additional costs. The key, therefore, is for your measurements to be as "non-intrusive" as possible. You don't want to be burdened too much by the overhead. Your measurements should be just enough.

What's especially important for people to figure out is what to measure. After that, how to interpret it. In radiology, for instance, you occasionally have neophytes who get the readings wrong. Even if they have a good MRI or a good set of work numbers,

they will misinterpret what it is saying. That is as much a risk in software as it is in medicine.

CAI: Are there any rules of thumb regarding what, specifically, to measure? More importantly, are there any rules of thumb about interpreting the data, making sense of the analysis process, and – in the end – developing the correct plan of action?

MICHAEL MAH: The SEI has produced good guidance on what these minimum measurements should consist of. Larry Putnam, in his book *Five Core Metrics*, also writes a lot on the subject. I have personally written about this extensively for the Cutter Consortium, where I serve as a Director.

The four core metrics are: 1) the amount of time teams spend working together; 2) the amount of work effort expended by these teams; 3) the amount of functionality needed to satisfy the client or end user; and 4) the degree of reliability or quality that exists. Larry has added a fifth metric – a productivity metric - which can be derived by examining the interrelationship between the other four metrics.

Going back to our medical metaphor, a doctor who is administering blood tests for cholesterol needs to know about LDL, HDL and triglyceride levels. He must also be aware of the minimum values and the range of normalcy. Furthermore, the doctor must understand how each of these components is interconnected.

The doctor is trying to collect data in order to figure out what needs to be fixed. He must then come up with a plan to fix it. In this respect, determining what to measure is obviously quite important. Moreover, one needs special training in order to read the data and interpret it correctly.

The software industry is replete with “neophyte radiologists;” that is, people in the field of software metrics who have not had the opportunity to acquire enough knowledge or experience to properly read and interpret their own metrics. Software measurement analysis is not particularly difficult, but many software developers continue to misread the data. Therefore, even if they have a “good MRI” or a good set of numbers to work

with, they can easily misinterpret what the numbers are indicating and wind up making choices that could constitute a "misdiagnosis."

It is also quite easy, especially for those of us in the West, to reduce things to piece parts and to look at them in isolation. Consequently, it is common for people to look at productivity in terms of functionality per unit cost or code per person month or function points per day. But they will only be looking at one or maybe two of the dimensions. In fact, there might be three, four, or five dimensions that need to be examined.

I often ask people how much schedule compression would take place if one were to double the staff. The fact is, if you're lucky, doubling the staff might reduce your time by maybe two months out of 10. So if we are racing to beat the clock and we add on an army of people, what happens then? Will bugs go down? When I ask people this in my seminars, no one says "Yes." Will bugs go up? When I ask this question everybody says "Yes." But here's the thing that shocks people: the bugs will go up by the square of the team size. That means if you double your staff you can expect bugs to go up by a factor of four or five.

My point is that these metrics are interconnected. There is a causal relationship. If you compress the date, there's going to be a predictable effect on defects. If you extend out the time and use a smaller team, you will discover that you have a mechanism for reducing the cost.

Everybody in our industry right now is focused on outsourcing. Everybody is focused on outsourcing because of the wage differentials, because it is less expensive for a programmer in India to code than it is for a Westerner. But there are times when the difficulty of cross-continental communication can impede the efficient and effective flow of information. This will result in a higher level of defects. And when you see the costs of trying to fix two and half times the number of bugs, which is what we're seeing from a recent sample of studies we've been conducting on offshore projects, you essentially wind up with a lot of hidden costs. That should come as no surprise.

So this data is interconnected. It's interconnected in a non-linear way. And people reading the numbers incorrectly mind wind up fooling themselves or misreading information, and this might cause them to make bad decisions, as in the case of the

outsourcing example. However, the good news is that this stuff can be taught, people can learn how to do it and do it right.

CAI: Do tools play a significant part in getting this right, or do you think that a measurement program can be reasonably successful without a heavy investment in tools?

MICHAEL MAH: A tool will only be as good as the accuracy of the underlying research. If the research is sound, the tool can be a very powerful diagnostic instrument. If the research is not sound, the tool will have little practical value.

Before I ever worked for QSM, when I first began searching for tools, I spent some time analyzing the underlying research behind them. When SLIM was in its very early stages, I was fortunate enough to meet Larry Putnam and had an opportunity to examine his work closely. Thanks to my strong physics background, as well as my fluency in mathematical algorithms, I was able to follow his research step by step. And I found it to be very sound.

In contrast, when I examined the published research of others, I saw a lot of linear math. Some of the concepts embodied in such math, such as function point per work month, simply did not take into account that we live in a non-linear world. By extending their logic, one would be able to cut a schedule in half by doubling the staff. But I knew that this was not accurate. A baby cannot be born in one month with the assistance of nine women.

In short, I think a tool can be good or not so good depending on the soundness of the underlying research. If it is sound, it can be a very powerful diagnostic instrument.

CAI: All of this might seem like a lot to take on especially for those organizations that are in Level 1 state. How would your advice differ for such organizations? Additionally, how would your advice differ for small organizations as opposed to large organizations?

MICHAEL MAH: There are a number of ways companies can jumpstart their ability to set more reliable dates, staff more reliably and estimate better. Simply getting a hold of some basic numbers is an essential practice that is not very difficult at all.

When I visit organizations at the cusp of trying to improve their maturity, I will always draw a picture of one of their projects on the white board. In the course of a half-hour, they manage to explain all of the various details about the project: how and when people joined the project; whether the peak was at the beginning, middle or end; how many program modules, objects or classes they were able to produce; how they tested for bugs and how many they found. These are all simple facts that any ground level organization should be able to produce. It is only a matter of mining the information and writing it down.

There are easy ways to read all of this, too. There are ways to chart a group of these projects and to draw a line right through the data points, e.g. projects that were small took this long, medium ones took that long, large ones took this long, etc. You can draw such lines between your data points and come up with an average schedule for how your projects tend to behave. Suddenly you can see trends. You can look at the same kind of chart for how many people you intend to staff on a project, or how much effort you intend to expend, or how many bugs you intend to find in code. This is all easy stuff to do. It's the kind of thing that my children are learning how to do in sixth grade algebra.

CAI: For people who are interested in learning more about these subject, is there any additional reading material that you might be able to recommend?

MICHAEL MAH: On the QSM website (www.qsma.com) there are many sample articles and reports that can be viewed and downloaded. Much of my own material can be found out there in our resource library.

Beyond QSM resources, I have often recommended *Waltzing with Bears*, which was written by Tim Lister and Tom DeMarco. Another great book is *Measuring and Managing Performance in Organizations*, which was written by Rob Austin. I can wholeheartedly recommend Larry Putnum's most recent book *Five Core Metrics*.

A book I am currently writing is *Optimal Friction: People Dynamics at Work in the Information Age*. Some of the ideas from this book are posted out on www.optimalfriction.com. The book examines the usage and effect of deadlines. In short, the energy created by time pressures can either breakdown the teamwork within a company or between two partner organizations, or it can become a sweet spot where just enough pressure is being applied.

In all organizations, even those outside the technology realm, people are always trying to create positive change. *The Tipping Point* by Malcolm Gladwell and his follow-up book *Blink* are books that apply to all organizations that are changing. I would recommend those as well.

Anybody interested in this material will also find a great resource in the Cutter Consortium website (www.cutter.com). Cutter is a think tank that hangs out in virtual space. The group consists of some of the most knowledgeable people in our industry. There is a lot of information and research out there.

The great thing about Cutter, though, is that is that you can always pick up the phone and say, "Can I arrange for Ken Orr, Michael Mah and Tim Lister to come down to my office for a consultation." It's kind of like one-stop shopping in that sense.

These are just a few good places for people to get started. In addition, readers should feel free to e-mail me directly at Michael.mah@qsma.com. If they do so, I will be happy to recommend other works that might extend or complement some of these suggestions.

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