

Kathleen K. Paige
President
Mark India, LLC
2758 S. Ives St.
Arlington, VA 22202-2311
V/Fx: 703-842-4312
kate.paige@markindiallc.com

"The Devil Does Power-Point"

- Aahhh..."Practical"; "Systems"; and "Measurement".
- 3 words that bring music to my ears.
- and talk about a timely topic....
- the defense news is rife recently w/ articles on the need to better understand and execute the development of software intensive systems.
 - <"Army Combat System May Cost \$307B, Pentagon Says"-.....>
 - <"Panel: Defense Software Problems Put Info Dominance 'At Risk'">
 - <"Etter Memo on Software Process Improvement"-.....>
- You, and especially the organizers of this Conference, are shining lights who are helping to lead the way to predictable, repeatable, and effective software development w/in the DoD. I applaud you, and I'm honored to be here w/ you.
- I do have a concern, however:
 - it is the devil that is in the details-
 - causing havoc if you don't chase him down in every minute detail;
 - acting as a decoy if you do chase him there, teasing your eye away from the big picture.
 - Two symptoms of the the "devil as decoy" syndrome are:
 - Slick Power-point presentations w/ glowing claims and adjectives, but a paucity of empirical data or experience.
 - (Remember: my vu-graphs will always be better than your system)
 - well-intentioned but blindered focus on process as an end in itself.
- Software and Systems measurement can be a powerful life-line to Managers of complex system developments, if it is tightly and explicitly connected to 1st Principles, and to context.
- Having been raised professionally as a systems engineer, w/ roots in software engineering, I view systems engineering as the discipline of finding practical solutions to real-world problems.
- The real-world problem to be solved, & the practical approach to it's solution, require adherence to the 1st principles--or fundamentals-- of engineering, leadership, and management; they also provide the context to measurement.

- It is in this context that I'd like to propose some, perhaps non-traditional, measures of software and systems for you to consider, if you haven't already.

- But 1st, to 1st principles;

- carefully selected, collected and analyzed data are the bedrock of engineering. But measurement is a means to an end, not an end in itself.

- on the other hand, it is also just the beginning. Measures tell little, if anything, in isolation. The questions they beg, and the trends they reveal over time, are what matter. IF....

- and it's a big IF.

-- If the parameters being measured are ones that will yield insightful, actionable information that, if acted upon, will improve the cost, speed or performance characteristics of the program. Otherwise, at best we are harassing the developers; at worse, we are leading the program down a road to Abilene, incurring cost and speed impact along the way. (It's a bit like the basic tenet of the Hippocratic oath: "1st, Do no harm".)

-- And of course as you well know, different measures are appropriate for different levels and stages of execution and management. For senior managers, less is usually more...if it's the right "less". That's the 'dashboard' concept you know well as a powerful tool for predicting the health and status of the program, providing diagnostic pointers to areas of potential weakness or illness.

-- The other big IF: If we are building the wrong product to begin with, or the requirements are vague, ambiguous, or really not well understood by the customer, no amount or quality of measurement will save the day.

- You all have the experts on the 1st "IF" here at this conference.

I'd like to spend a little time talking about the 2nd "If" :

-- building the right thing, to the "right" requirements, starting w/ an example from my background in Air & Missile Defense, where the 'End-in Mind' is "Ordnance on Target".

- The Aegis combat system has used 5 key performance parameters, or Cornerstones, derived from the Withington Study conducted in 1965 by an Admiral recalled to active duty by the Secretary of the Navy to address the serious problems in the guided missile fleet of the day, and to propose the architecture for the future Surface Air Defense system that would become Aegis.

--<discuss the cornerstones>

--these cornerstones have provided the unifying and guiding themes for the program for nearly 40 years, and remain as relevant today as they were when proposed by RADM Withington.

-- Does Aegis have more traditional specifications? Of course they do. Engineers need specs. But the program always remembers that requirements are nothing more nor less than decisions; that there are no perfect specs or requirements; and they are best when iterated among the engineers, the program manager and the operators, being refined, edited or modified as each group learns more as the development process proceeds through its paces.

-- This topic is addressed well for IT systems by Michael Schrage, the co-director of the MIT Media Lab's eMarkets Initiative and an instructor on innovation economics for several MIT executive education programs.

---<quotes from "Never go to a client...."article>

- This brings me to the first thought I would ask you to ponder:

-- How can we measure the effectiveness of the reqmnts definition process? e.g.

--- can the reqmnts be rolled up, dashboard fashion, into 3-5 key performance parameters, or Cornerstones, that provide the long-term guiding vision?

---can the iterative prototype (or specification evolution) approach be used to advantage in the development of complex weapon or information dominance systems?

If so, how to measure the effectiveness of each iteration? The point at which more iterations provide diminishing returns?

- My next challenge to you is how to measure the robustness of software--i.e. the ability of the system as implemented in a software intensive system to fight through adversity and still successfully fulfill it's mission?

This is an imperative quality for weapon systems.

--<use example of SM-3 IR seeker during FM-6>

- My last question for you to ponder is how to best measure the effectiveness of the PSM Transitions. You may well have the answer to this question already, because you likely practice what you preach. But even if you do have the answer, I would propose that you exploit the power of such information, disseminating the data showing the extent to which PSM techniques have helped contributed to containing or improving the cost, sked, quality and/or customer satisfaction of those programs that have incorporated PSM disciplines into their processes.

-DoD is looking for life-lines to save it from software development Hades.

Measurement of software & systems, practically done, always w/ the end-in-mind, is the core of that lifeline.

- Each of you, effectively measuring the right things throughout the development of software intensive systems, are part of keeping the devil at bay.

Thank you.