Through a Glass, Darkly
Separated by a Common Language, Version 2

Robert L Glass

Awhile back, I titled my Through a Glass, Darkly column “Divided by a Common Language.” Although that column led off with a brief discussion of US vs. Australian language differences (“separated [not ‘divided,’ I hasten to correct] by a common language” is what George Bernard Shaw said about England and the US a number of years ago), the essence of that column was about terminology differences between the Information Systems and Computer Science / Software Engineering fields. “Requirements triage,” which I asserted could be a legitimate interest of Systems Analysts in the IS field, was the specific topic. I borrowed it from a software engineering book by Alan Davis called Just Enough Requirements Management.

I’d like to return to that notion of the US and Australia language differences. There are some real shockers, I have been discovering in the nearly four years I have been resident in Australia, after a lifetime in the US. Uses of language in Australia that are 180 degrees different in meaning from the same expression in the US. Let me explain.

Suppose you are at an IS project meeting, there is a discussion of an important new direction for the project to take, and the boss says “let’s punt on that.” What does the boss mean? Is it
a. let’s give up on this idea
b. let’s move forward on this idea, in fact betting on it
c. who ever thought up this crazy idea, let’s fire him or her?

Here’s the thing. If you’re in the US, (a) is the correct interpretation. This meaning of punt is derived from the all-American sport of football, where to punt is to turn the ball over to the other team, thus giving up control for a while.

If, on the other hand, you’re in Australia, (b) is the correct interpretation. This meaning of punt is derived from the all-Australian sport of betting, where to punt is to place a bet (on a team or a horse, for example), where the expectation is you think that team or horse will be a winner.

(Fortunately, I know of nowhere where the third meaning, (c), holds).

Let’s play this game again. Suppose you are at an IS project meeting, there is a discussion of an important new direction for the project to take, and the boss says “let’s table that.” What does the boss mean? Is it
a. let’s postpone further discussion on this
b. let’s put this on the table for a decision to be made
c. who ever thought up this crazy idea, let’s fire him or her?
Here’s the thing, again. If you’re in the US, (a) is, once again, the correct interpretation. It’s not quite as clear where this meaning comes from. But there is no doubt, in the US, that tabling something means to defer it.

If, on the other hand, you’re in Australia, (b) is the correct interpretation. It’s no more clear where the Australians got this meaning than where the Americans got theirs, but – once again – there is no doubt that tabling something means to place it in position for further discussion.

(Fortunately, yet again, I know of nowhere where the third meaning, (c), holds).

These are, of course, not the only 180 degree language differences between the US and Australia (and England, which is where most Australian definitions come from). The word “pissed,” for example, means “angry” (when followed by “off”) in the US; in Australia, it means drunk. And the word “rubber,” which has a diversity of meanings in the US (including tires, wet weather footwear, the card game bridge, and – most famously – a certain sexual artifact), in Australia it means “eraser.” (Brits and Aussies report some interesting, perhaps even hilarious, reactions when they use the British term in a US setting!)

And then there’s the world of sport, where in rugby a score is a “try” (that gives whole new meaning to the old saying “if at first you don’t succeed, try, try again”!), and in cricket the most important matches are “tests’ (whereas to an American a test is a lesser form of examination).

But, fortunately, none of these other words it likely to come up very often in an IS project setting!

Acknowledgement: I wish to thank Phillip Armour (a self-confessed “whinging POM” (*) now living in the US), for some interesting exchanges on this topic.

* - “Whinging,” an Australian term, means “to complain frequently” (as in “whining”). “POM” is a somewhat-derisive Aussie term for a British expat (it is an acronym derived from “Prisoner of Mother England” or “Prisoner of her Majesty” which was the status of most early Australian Caucasian residents).

“Through a Glass, Darkly,” is a Biblical expression for the unclear way in which we see the world around us.

Robert L. Glass is president of Computing Trends, publishers of The Software Practitioner newsletter, and an Honorary Professor of Software Engineering at Griffith
University, Brisbane, Australia. He has been active in the field of computing and software for over 50 years, largely in industry (1954-1982 and 1988-2005), but also as an academic (1982-1988 and 2005-present). He is the author of over 25 books and 90 papers on computing subjects, Editor of The Software Practitioner newsletter, Editor Emeritus of Elsevier’s Journal of Systems and Software, and a columnist for several periodicals including IEEE Software (“The Loyal Opposition” column). He was for 15 years a Lecturer for the ACM, and was named a Fellow of the ACM in 1998. He received an honorary Ph.D. from Linkoping University in Sweden in 1995. He describes himself by saying “my head is in the academic area of computing, but my heart is in its practice.”