Finding local lessons in software engineering

Tim Menzies
West Virginia University

Tuesday, February 23, 2010
3:00pm Rm. 110 Purdy-Kresge Library

Abstract:

Where are the general principles of software engineering that we enshrine as requirements, teach our students as gospel, and enforce in legislation? Despite decades of work, there are surprisingly few unequivocal empirical results where some effect was observed in one software project then, when applied to another, the same effect repeated. Should we expect such general principles? Or is intelligent software engineering really a process of finding the best local lessons for particular projects?

This talk explores the generality of lessons learned in one project, across many others. A simple repeatable experiment is defined (using case-based reasoning) where is shown that what is best for project A is irrelevant to project B. The experiment is not conclusive, nor is it intended to be. Rather, it is intended to offer a baseline result which, hopefully, will be improved on in the future.

Biography:

Dr. Tim Menzies has been searching, with limited success, for general principles of software engineering for two decades. He has been working on advanced modeling and AI since 1986. He received his PhD from the University of New South Wales, Sydney, Australia and is the author of 190 referred papers. A former research chair for NASA, Dr. Menzies is now an associate professor at the West Virginia University's Lane Department of Computer Science and Electrical Engineering. For more information, visit his web page at http://menzies.us or view his papers at http://menzies.us/papers.php.