A Program Understanding Theory of Software Complexity

Many complexity metrics have been proposed to help predict the cost and effort of producing new software systems. However, more than 80% of the effort associated with a software project is not in its initial development, but rather in its evolution and maintenance. Existing complexity metrics have been shown to be more or less isomorphic to simple code size, and are of little help in predicting potential maintenance effort, other than observing that bigger systems require more work.

In this talk I will step back and ask the question "what makes software maintenance difficult?" Beginning with a reminder of the limitations of programmers as human beings and drawing from observations from software maintenance research, we will explore the known and unknown causes of difficulty in software maintenance in search of a feature-based metric that can capture what really makes maintenance hard. Surprisingly, we will discover that it all boils down to one thing: there is no problem in software maintenance that cannot be made worse by another level of indirection.

Based on this observation, I will conclude with a concrete proposal for a new kind of feature-based software complexity metric, and demonstrate its application to estimating the complexity of "make"-based build systems.

About the Speaker

Dr. James R. Cordy is Professor and past Director of the School of Computing at Queen's University in Kingston, Ontario, Canada. From 1995 to 2000 he was Vice President and Chief Research Scientist at Legasys Corporation, a software technology company specializing in legacy software system analysis and renovation. As leader of the TXL source transformation project with hundreds of academic and industrial users
worldwide, he has been involved in software analysis and transformation systems for more than 25 years. He has published more than 180 refereed contributions in software engineering, programming languages and artificial intelligence.

Dr. Cordy serves widely as member and chair of conferences and workshops in programming languages and software engineering, recently co-chairing the IEEE 2011 International Conference on Software Maintenance, the IEEE 2012 International Working Conference on Source Code Analysis and Manipulation, the 2012 International Working Conference on Reverse Engineering, the 2012 Dagstuhl Workshop on Software Clone Management in Industrial Application, and the 2013 IBM CAS International Conference on Computer Science and Software Engineering. He is an ACM Distinguished Scientist, a Senior Member of the IEEE, and an IBM CAS Faculty Fellow.