NATIONAL UNIVERSITY OF SINGAPORE

School of Computing

PH.D DEFENCE - PUBLIC SEMINAR

Title:	Automated regression testing and verification of complex code changes
Speaker:	Mr Marcel Boehme
Date/Time:	26 June 2014, Thursday, 10:00 AM to 11:30 AM
Venue:	Executive Classroom, COM2-04-02
Supervisor :	Dr Abhik Roychoudhury, Associate Professor, School of Computing

Abstract:

Software changes constantly. For instance, the Linux operating system has been evolving over the last twenty years to a massive 300 million lines of code and each day an enormous 16 thousand lines of code are changed in the Linux kernel alone!

How can we check these software changes cost-effectively? Even if we are confident that the earlier version works correctly, changes to the software are a definite source of potential incorrectness. The developer translates the intended semantic changes of the program s behavior into syntactic changes of the program s source code and starts implementing the changes. Arguably, as these syntactic changes become more complex, the developer may have more difficulty understanding the semantic impact of these syntactic changes onto the program s behavior and how these changes propagate through the source code. Eventually, the syntactic changes may yield some unintended semantic changes. Existing program functionality that used to work may not anymore. The result of such unintended semantic changes is software regression.

In this dissertation we put forward the following thesis: The correctness of a complex source code change can only be checked cost-effectively by accounting for the interaction among its constitutent changes. In other words, it is not sufficient to exercise each constitutent change individually. In particular, we present two approaches to checking the correctness complex source code changes, one improving the efficiency of a very effective technique and one improving the effectiveness of a very efficient technique. We also formally define and investigate a new class of regression errors that arise from the interaction among the constituent changes. We introduce change complexity as a measure of the interaction among the constituent changes based on which we define and study the complexity of regression errors.

In summary, we answer how to determine the semantic impact of a complex change and just how complex a complex change really is. We answer whether the interaction of the simple changes constituting the complex change can result in regression errors, what the prevalence and nature of such (change interaction) errors is, and how to expose them. We answer how complex a complex error really is and whether regression errors due to change interaction are more complex than other regression errors. We make available a benchmark suite, consisting of 70 actual regression errors and tools that measure change and error complexity and that generate test cases to expose change interaction errors.