Title: Improving the quality of software using testing and fault prediction

Abstract: Software will always have bugs, and as software continues to become a more and more pervasive part of our lives, software bugs will continue to affect more people than ever. Our confidence in software systems depends on our confidence in the exhaustiveness of our testing. As software systems get more complex, the task of exhaustive testing becomes more complex and even infeasible in some cases. In order to build less error prone systems, we therefore need to not only focus on quickly and efficiently identifying bugs through testing and verification of software, but also on identifying factors associated with bugs in order to prevent them in the first place. Software development is a complex process that requires coordination between individuals and technology, and as we look to predict and prevent bugs, we need to examine the effect that socio-technical factors have on code quality. In this talk, I present my research on identifying socio-technical factors that are associated with software bugs by using a combination of mining software repository and machine learning.