Máster en Ciencias y Tecnologías de la Computación

Doctorado en Ciencias y Tecnologías de la Computación para Smart Cities

Curso 2014-15

Seminarios de Investigación

1er seminario:

Título: Design Rule Spaces A New Form of Architecture Insight

Conferenciante: Rick Kazman

Fecha: 26 de noviembre de 2014

Lugar: Sala de Grados (ETS de Ingeniería de Sistemas Informáticos)

Hora: 19:00

Descripción: Locating and fixing bugs is the most expensive maintenance activity over the lifetime of the vast majority of software projects. Numerous techniques have been proposed to aid in locating buggy files--from model checking to metrics-based prediction, evolution history-based prediction, and so on. However none of these has been highly successful. We believe that the reason for this persistent problem is that the architectural relationship among defective files has never been explored. Our recent research has shown that, in large-scale software systems, defective files seldom exist alone. On the contrary, defective files are usually architecturally connected, and their architectural structures exhibit significant design flaws that are the root causes of their bugginess. The implication is that only fixing individual defective files, as is done today, will not remove the root causes of bugginess or avoid the seemingly inevitable ripple effects of causing more bugs when fixing one. Removing the architecture roots of bugginess is, we claim, the key to reducing overall defect rates in a project. Our strategy is to discover the architecture roots that cause the bugginess in groups of files, to categorize recurring architecture issues into architectural anti-patterns, and to refactor these anti-patterns. We prosecute this strategy first by modeling software architecture using a novel representation called design rule space (DRSpace). A DRSpace can express software structure relation, quality, and evolutionary information simultaneously, making it possible to assess the relation between software architecture and defect rates. Second, we automatically extract defect-prone architecture roots by combining static architecture analysis with software revision history, so that the occurrence of such roots can be detected and monitored continuously through the software development process. Third, we can automatically discover architecture flaws in these

DRSpaces, and group these flaws into architecture anti-patterns. Fourth, we can propose refactoring strategies to remove architecture roots. Finally, we can build economic models of the before and after states, so that management stakeholders can make informed decisions about the costs and benefits of such refactoring activities.

Rick Kazman is a Professor at the University of Hawaii and a Principal Researcher at the Software Engineering Institute of Carnegie Mellon University. His primary research interests are software architecture, design and analysis tools, software visualization, and software engineering economics. He also has interests in human-computer interaction and information retrieval. Kazman has created several highly influential methods and tools for architecture analysis, including the SAAM (Software Architecture Analysis Method), the ATAM (Architecture Tradeoff Analysis Method), the CBAM (Cost-Benefit Analysis Method) and the Dali architecture reverse engineering tool. He is the author of over 150 peer-reviewed papers, and co-author of several books, including Software Architecture in Practice, Evaluating Software Architectures: Methods and Case Studies, and Ultra-Large-Scale Systems: The Software Challenge of the Future.

Madrid, a 24 de noviembre de 2014

Fdo.: D. Jesús García López de Lacalle
(Coordinador de la asignatura)