A Practical Software Measurement Mechanism

Allen P. Nikora
Jet Propulsion Laboratory,
California Institute of Technology
Pasadena, CA 91109-8099
Allen.P.Nikora@jpl.nasa.gov

John C. Munson
Computer Science Department
University of Idaho
Moscow, ID 83844-1010
jmunson@cs.uidaho.edu

Abstract

Over the past several years, we have identified relationships between measurements of a software system's structural evolution and the number of faults inserted between subsequent versions of that system. This makes it possible to estimate the total number of faults inserted into a system during its development. Developers and testers can use this information to identify those modules having the most faults and allocate their resources accordingly. A practical structural measurement system has to be developed and integrated into the software development environment. The system must have the following characteristics:

- Measurements must be meaningful and repeatable.
- Measurements must be consistent.

We describe a measurement capability we have implemented at the Jet Propulsion Laboratory, consisting of three components:

- Structural measurement – identifies modules that have changed since the last set of measurements were taken, measures them against a baseline, and computes their fault indices.
- Fault burden computation –computes each module's proportional or absolute fault burden from the fault indices.
- Fault measurement and identification – for each fault repaired, determine the point at which it was inserted into the system. This information is used to develop a model from which absolute fault burdens can be estimated.