Issue Date: 06-APR-2015

MC/DC TESTING COST AND RISK REDUCTION

As explained at Wikipedia ("http://en.wikipedia.org/wiki/Modified_condition/decision_coverage"), MC/DC testing is a form of structural test coverage evaluation. It can be performed on source code in order to ensure adherence to important safety-critical philosophies such as: [1] there is no "dead" (i.e., unexercisable) code; [2] there are no "order of boolean operation" dependencies in the contained logic (i.e., the code is deterministic when evaluating Boolean expressions); and [3] there are no self-conflictual Boolean statements (e.g., A and B and NOT C and... and NOT A). Note that, in general, structural test coverage is just one aspect of overall unit testing and that, unlike the functional testing aspects most closely associated with unit testing, structural test coverage is not necessarily associated` with the actual functionality of the source code under test.

Of the various types of structural test coverage that can be performed, MC/DC testing is the most stringent and, therefore, the most time consuming -- by far. It is required by both RTCA/DO-178B and RTCA/DO-178C for all software that has a safety-criticality designation of Level "A" ("Catastrophic"). Note that the types of structural test coverage that are required for the less safety-critical RTCA/DO-178 designations are the following, where each of which is typically one order of magnitude less time- and resource-consuming than the one just above it: [a] Decision coverage testing is required for Level "B" ("Hazardous"); and [b] Statement coverage testing is required for Level "C" ("Major"). Additionally, note that no structural test coverage evaluation is required for the remaining levels (i.e., Levels "D" or "E").

It is typical to purchase a third-party tool (such as VectorCAST by Vector Software, TestMate by IBM/Rational, AdaTest by QASystems, etc.) in support of MC/DC testing. Additionally, typical MC/DC testing consumes a minimum of one (1) man-year to complete the covering of 100 KSLOC of source code. However, the actual amount of this labor varies a great deal, since this value is highly dependent upon the construction of the source code; the effort can easily be increased by perhaps 250% or decreased by perhaps 75%. The difference is in the use of smart software design. In general, this value increases perhaps quadratically with increases to the code's cyclomatic complexity and increases in the size of the various composite Boolean predicates. Furthermore, if (when!) source code rework is found to be required due to coverage failures that result from MC/DC testing, yet another source of overhead is added. This overhead can result in an additional 10% to 20% delivery delay.

MC/DC testing will be a burden. To ease this burden, you will want to integrate time-saving strategies before project start. Assurant Design Automation (ADA) offers unique solutions to this challenge and, coupled with our MC/DC experience, can help the 580th SMXS team fully incorporate them seamlessly, with no additional overhead, and in ways that are re-usable throughout the project's lifetime.