Improving Software FMEA Processes and Results

System reliability and safety can be improved by applying failure modes and effects analysis (FMEA) to system software as well as hardware. Software FMEA, like its hardware counterpart, can be performed at either a detailed or high level. A detailed analysis typically examines variables at the source code level and seeks to determine system effects when any variable assumes any unexpected value – a difficult and tedious undertaking. Worse, the analysis cannot be started until late in the program when at least parts of the software have been coded. In contrast, a higher level analysis might examine messages exchanged among system CSCIs and typically seeks to determine system effects when any part of message has an unexpected value. This kind of analysis can be started sooner since messages are often defined by interface control documents (ICDs) long before source code even exists.

A detailed analysis is the most costly and is usually limited to critical system areas. However, it generally gives the most complete and accurate results since it is based on source code – the closest representation of actual system operation. Not surprisingly, a high-level analysis requires less effort and can be economically provided to all parts of a system. It also can reveal problems at the detailed level because many kinds of low-level failures cause unexpected message contents and timing. However, the high-level analysis usually provides less complete and accurate results because the ICDs on which it is based usually reflect designer intent rather than actual execution.

Fortunately, the tedious detailed FMEA can be automated in part and the higher-level FMEA can be streamlined without automation. The proposed presentation explores practical considerations and techniques developed while performing real-world analyses that improve process effectiveness and results for analyses at both levels. Topics include development of failure modes, using databases to minimize human effort and maintain consistency, avoiding unnecessary details in results, development of ground rules and assumptions, and ways of proceeding when information isn't available.

Nat Ozarin is a senior engineering consultant at The Omnicon Group Inc. (www.omnicongroup.com), a company specializing in reliability and safety analysis for the military, medical, industrial, and transportation industries. His background includes hardware engineering, software engineering, systems engineering, programming, and reliability engineering. He received a BSEE from Lehigh University, an MSEE from Polytechnic University of New York, and an MBA from Long Island University. He is an IEEE member and was named Reliability Engineer of the Year by the IEEE Reliability Society in 2009.
