Medical software standards: Boon & bane

Mark Pitchford - October 01, 2013

The medical device industry has lagged other industries in the adoption of safety standards for software design. The situation has improved with the introduction of IEC 62304, and medical device software reliability has improved. But there may be a cost to innovation.

Recent quality concerns are driving many industries to start looking seriously at ways to improve the quality of software development. Not surprisingly, there are marked differences in the quality of software in the different sectors. The automotive industry, for instance, does a good job of listing all software requirements in a database and was the original of the MISRA software coding standards. The railway and process industries have long had standards governing the entire development cycle of electrical, electronic and programmable electronic systems, including the need to track all requirements. The automotive sector, though, has only recently introduced a standard for a similarly broad-based approach.

By contrast, medical software guidelines are typically more in line with low-risk applications, even when a software failure could result in serious physical injury or death. Despite this apparent disparity between medical software and other safety critical standards, the case for software, which has been proven to be reliable through standards compliance and requirements traceable process, is becoming ever more compelling.

The US government is well aware of the incongruence of this situation and considered ways to counter it. The Drug and Device Accountability Act was one such attempt, although it died in committee. Still, manufacturers are being held accountable. Recently, the FDA took punitive action against Baxter Healthcare and their infusion pumps, which the FDA has forced the company to recall.

The net result is that many medical device providers are being driven to improve their software development processes.

This drive for improvement is underpinned by the IEC 62304 standard which, like other standards in the Railway, Automotive, Nuclear, and Process industries, is based on the more generic IEC 61508 standard.
Perhaps is it surprising – alarming, even – that medical device software is only now being brought in to line with best-practices established eight or more years previously in other industries. However, perhaps a clue as to why this might be can be found in the classification system applied to medical device software.

A common concept in the standards applied in the safety critical sectors is the use of a tiered, risk-based approach for determining the criticality of each function within the system under development. Typically known as “Safety Integrity Levels” (SIL), there are usually four or five grades used to specify the necessary safety measures to avoid an unreasonable residual risk of the whole system or a system component. The SIL is assigned based on the risk of a hazardous event occurring based on the frequency of the situation, the impact of possible damage, and the extent to which the situation can be controlled or managed.

But in medical systems, the SILs are more aligned to the FDA’s pre-existing safety classifications than to probabilities. In IEC 62304, the safety levels go from the lowest Class A (No injury or damage to health possible) to the highest Class C (Death or serious injury possible).

IEC 62304 also differs from other similar standards in that the classification has to apply to the whole system. Other standards, such as ISO 26262, permit less critical parts of the system to comply with a less demanding SIL than the more critical functions.

In short, IEC 62304 was designed to combine the existing best-practices of the medical device
industry with the proven success of IEC 61508 and its derivatives. It is extensive and comprehensive as a result.

So, that’s all good news. Medical device software is falling in line with standards elsewhere and we can all sleep easier in our hospital beds when we are plugged in to Monty Python’s "machine that goes 'ping'."

But there is a downside. Medical device software is frequently developed by very small startup companies, perhaps more so than any other safety-critical sector. In the other sectors, the overheads associated with standards such as these are mostly carried by large organizations. As the overhead involved with the transition from prototype to production can be daunting, the standards can create a significant barrier for small companies to overcome.

How many potentially life-saving devices will never see the light of day as a result?

**Also see:**

- Lessons learned: using formal methods to develop medical device software
- Focusing on Traceability in Software Development for Safe Medical Devices
- Which OS for IEC 62304 Medical Devices? (webinar)
- Which OS for IEC 62304 Medical Systems? (paper)
- IEC 62304 Compliance: Recognizing Risks and Closing the Gaps
- Using Dynamic Software Analysis to Support Medical Device Approval