SupplyChainDigest™
Your First Stop for Supply Chain Information

Getting Software Embedded in Manufactured Products Right

Too Many Manufacturers Use Traditional Engineering Approaches to Embedded Software Engineering; Reducing Complexity

SCDigest Editorial Staff

Software accounts for more and more of the total value of a growing array of manufactured products, and manufacturers are in some cases adding software developers faster than even the largest traditional software companies. But it has proven tough for many manufacturers to get their software development right, leading to high costs, customer problems and recalls.

The growth of software and related technologies in manufactured products has been remarkable. This trend has been bolstered by improved technology that makes it possible to replace existing mechanical functions, and manufacturers looking for ways to differentiate their products and achieve higher margins. While software development can be expensive, the variable cost per unit can be quite low.

In the early 2000s, the CEO of auto parts maker Eaton Corp. famously said in an interview that if an Eaton product "didn't have a chip in it," he wanted to either redesign the product or get out of the business.

Many other manufacturers seem to agree, and now the emphasis is switching from chips to more direct software development.

German industrial giant Siemens, for example, reportedly now employs more software engineers in its high-tech businesses than do large software companies such as Microsoft, Oracle, or SAP.

But the trend is not without problems. Software bugs have been responsible for a high number of the recalls in the automotive industry. Cars are now controlled by dozens of electronic control units (ECUs) that are networked to the OEMs usually proprietary software systems and inboard computers – and if something doesn't work well with that software, the costs can be huge. More and more manufacturers will face similar risks.

German industrial giant Siemens, for example, reportedly now employs more software engineers in its high-tech businesses than do large software companies such as Microsoft, Oracle, or SAP.

Best Practices for Embedded Software

A recent article in The McKinsey Quarterly by **Wolfgang Huhn** and **Marcus Schaper** offers a number of best practices for embedded software development. While focused largely on the automotive supply chain, the recommendations seem equally applicable to any manufacturer looking to improve the productivity, quality and value from internal embedded software development.

Reduce the Complexity of Features: Complexity in the software adds tremendously both to initial and aftermarket costs. The development time is not linear as the number of features goes up – it lengthens exponentially. Doubling the features will add way more than double the total effort.

Rigorously analyze the value of proposed new software features, and cut those that don't add up – or which won't really help sell more product.

Adopt a More Mature Software Architecture:

Getting Software Embedded in Manufactured Products Right (Con't)

While good software development relies on a modular and smartly layered technology stack, for many manufacturers (and traditional software developers too), short cuts or poor design often lead to "spaghetti code" that becomes increasingly difficult to maintain and develop upon.

"Our research finds that the architecture of embedded software is one of the weakest spots of its development, lagging half a grade behind the development of comparable traditional software," the McKinsey authors say. The key lesson: pay close attention to architecture decisions, and be wary of short cuts to save time today but that may be very costly down the road. Also look hard at the whether there is real value in proprietary platforms.

Understand the Economics: Manufacturers often don't fully recognize the full lifecycle costs of the software. They will avoid lower quality traditional components in their products that would cause large customer satisfaction or warranty cost issues, but then take shortcuts in software development or especially testing that have the same impact in the end. The same can be true for the electronic components associated with the hardware.

"When suppliers and OEMs undertake design-to-cost planning, they must learn to consider a broader set of criteria, including the lifetime costs of complex software and less expensive hardware," the authors say.

Improve Development Processes: The traditional approaches that have worked for decades for many manufacturers in terms of physical products are often not well suited to embedded software development.

Software engineering and development is different, and embedded software development itself has important differences than traditional business software engineering.

Best practices, McKinsey says, include the use of cross-functional teams of experts (feature-based development), development methodologies that apply common models and simulation tools, and "time boxing" (giving developers specific deadlines for deliverables).

Changing practice and mindsets will not be easy, the authors say. However, embedded manufacturing software development needs "to shift from a hardware-development mentality to one more attuned to the iterative nature of software development," they note.