COTS Software: Reduce Risk and Time to Market for Safety-Critical Systems

Surveys have shown that safety-critical device developers can spend as much as 30% of total development time on their own custom runtime platform. This means that roughly one third of their development budget goes into software that, in most cases, isn't creating value for the company. Less time is spent on creating new innovative capabilities—less time on testing and less time making sure the product is more competitive.

Why aren’t these companies using commercial-off-the-shelf (COTS) software? In the past, a general-purpose OS, such as Microsoft Windows or Linux, wasn’t small enough and appropriate for the safety-critical market. Moreover, many commercial, embedded real-time operating systems (RTOSs) weren’t suitable because they lacked the appropriate testing or certification needed for use in safety-critical systems. However, this is no longer the case. There are very compelling reasons to consider COTS solutions (both hardware and software) for safety-critical systems.

The COTS Advantage
Leveraging COTS software in safety-critical systems provides various advantages over a home-built, custom solution. Consider the following:

- Commercial services and support: COTS products are maintained, serviced, and supported by commercial vendors. In contrast, home-built solutions require in-house maintenance and support.
- Certification and certification evidence: Certifications and certification evidence means that these reused components are tested and documented to the appropriate safety standards. This reduces the overall certification, testing, and documentation burden.
- Commercial tools and middleware: Embedded systems are getting more connected and more complex. The right tools and middleware greatly increase productivity in all parts of software development.
- New technology: Keeping a custom, home-built solution up to date on the latest hardware and software technology is time consuming and expensive. Commercial solutions provide a future roadmap of technological advancement. For example, consider embedded virtualization, multicore, wired and wireless networking, and graphics.

Certification and Certification Evidence
Most safety-critical systems require certifications to show their readiness for safe use in the marketplace. For aviation, it may be the RTCA DO-178B while for industrial, it could be IEC 61508. For medical, systems require the IEC 62304 or the FDA's Good Manufacturing Process (GMP or GAMP). In order to use COTS software in these devices, the software must generally be developed to the same stringent standards as the end products that it will run. Moreover, the vendors must supply the necessary certification evidence to prove that these strict standards were met. The certification evidence is a key deliverable for safety-critical COTS software. Without it, the product offers little benefit to the end customer. In some cases, such as with IEC 61508, the COTS software can carry a certification as well. For example, Wind River VxWorks Cert 6 has been certified to IEC 61508 Safety Integrity Level (SIL) 3.

More than just the OS
It’s important to note that the safety-critical device developer is responsible for whatever COTS software is used. In the end, the final device or system is certified—not the individual components. Re-using software in these situations has risks—both safety and technical—that must be mitigated. Developers considering COTS software need to take into account the scope of the certification or certification evidence.

Developers look to their COTS vendors to reduce this risk by supplying certification and certification evidence for the software they provide. Any COTS software that doesn’t have...
the necessary certification evidence is considerably more risky. From an RTOS perspective, it’s important that more than the basic operating-system functions are tested and documented. Any component, such as the file system or network stack, which is used without the necessary certification evidence must then be certified by the device developer. This increases the testing and documentation burden (reducing the value of COTS software).

Safety is a Team Effort

Using COTS software for safety-critical devices wouldn’t be viable if the vendor and developer didn’t work closely together. Although the value of COTS software is considerable on its own, the full advantages of the software won’t be achieved without the appropriate level of training, support, and engineering services to get the developer up and running and productive on a COTS software platform. Moreover, no two safety-critical systems are alike. Custom engineering is often needed to bridge the gap between COTS capabilities and system requirements. Therefore, developers need to look at the overall capabilities of COTS-solution partners from product to support.

The advantages to using COTS software for safety-critical systems are clear. Developers can greatly reduce their risk, time, and effort in product development. However, COTS software also needs to meet the stringent standards under which the device will ultimately operate. To get the most out of COTS software, developers need certified software (or components with certification evidence) as well as tools, middleware, and services and support to realize the maximum advantages.

Bill Graham is a software industry veteran with more than 20 years of management and development experience spanning embedded and real-time systems, UML modeling, and object-oriented design. He is a frequent contributor of ideas and insights about many aspects of device software design and development through speaking engagements, web seminars, blogs, and contributed articles. Graham holds bachelors and masters degrees in electrical engineering from Carleton University in Ottawa, Canada.