

# MDDI

Medical Device & Diagnostic Industry

## It's Time to Embrace New Software Technology

New platforms can change the way medical device companies design, upgrade, and write software for their devices.

**Andrew Dallas**

*Full Spectrum Software*

In the past three years, a revolutionary change in software has occurred that has been largely unnoticed by the medical device industry. This change will fundamentally alter the way in which software is designed, developed, and tested for medical devices. New application-level and embedded software technology enables designers to rethink how they design, upgrade, update, and write software for medical devices.

What is this change and why is it so significant? In short, the new Microsoft and Linux application and embedded systems platforms and associated tool sets allow developers to create major sets of functionality largely by integrating components that exist within a platform. This is a fundamentally different development process from the traditional process of writing lines of code to create the same functionality.

Other highly regulated industries such as military and aerospace have embraced the concept of buying pre-built systems and software components or commercial off-the-shelf (COTS) systems. COTS applies to



hardware as well as embedded and application-level software.

New Microsoft and Linux software technology has reached a point at which it is more cost-effective to substantially upgrade existing legacy software or even rewrite existing software rather than try to continue to patch or incrementally upgrade old software. Both platforms have made extraordinary advances in recent years. Both also have certain strengths and weaknesses, so, in reality, the choice of using one platform as opposed to the other is an engineering decision.

According to a report released by

market research firm Embedded Market Forecasters, embedded development projects based on Microsoft's Windows embedded operating system platforms (specifically, Windows CE.NET and Windows XP Embedded) are completed 43% faster and at 68% lower cost, on average, compared with similar projects using embedded Linux.

The study, titled "Total Cost of Development: A Comprehensive Cost Estimation Framework for Evaluating Embedded Development Platforms," derives its quantitative conclusions from a cost-based framework for comparing embedded operating system development alternatives that was developed by the report's author, Jerry Krasner.

Of course, the Linux community offers its own assessment. According to LynuxWorks Inc., a pioneer in the Linux embedded-systems industry, its embedded operating systems are based on open standards and are used in important products made for markets such as communications, aerospace and defense systems, medical devices, and automotive. The company claims:

The reliability requirements of embedded environments are far greater than those of desktop systems. It is common for embedded developers to design systems whose downtime is measured in seconds per year. Linux has achieved the degree of reliability required for embedded systems through a number of factors:

- Parametric testing—Linux has undergone a far greater degree of testing than Windows XP has due to active efforts by its users to isolate and repair bugs found in the many diverse environments and applications within which it has been deployed. This “many eyes” approach makes use of a technique common in distributed processing—parametric execution—in which the same code is executed with slightly different inputs. In the case of Linux, this form of testing is augmented by “white box testing,” the testers being able to peruse through the source code, identifying and, in many cases, repairing the bugs. The feedback mechanism implicit in the Linux community ensures that the significant bug fixes make it back to the central kernel group.
- Linux was designed around the highly refined UNIX architecture. UNIX has had 40 years to mature as both an API and kernel architecture. Windows NT, the OS base upon which Windows XP is based, is scarcely 10 years old. UNIX has been tested and proven portable and robust on countless CPU architectures where Windows has spent most of its life on the x86 architecture.

### **Slow to Adopt**

There are several valid reasons why the medical device industry has adopted software technology at a much slower pace than other industries. Making significant changes to either embedded or application-level software involves addressing regulatory guidelines, testing, and compliance issues. These constraints are legitimate. However, there are some unrelated historical reasons that software innovation proceeds at a slower pace in the medical device industry.

From a historical perspective, many well-established medical device companies began life as hardware vendors. The first version of the software may have been written by the original team of hardware designers or the scientists who needed to get a product out the

door. The code may have been written for DOS or another antiquated operating system. Some of this old code has been upgraded, with great trepidation, and only when absolutely necessary. Sometimes the databases and programming languages are so old that the tools they were written in are discontinued and unsupported. The time and expense of retesting and certifying the old software provides both a business rationale and a certain level of motivation to maintain the status quo.

From a marketing perspective, it is difficult, time-consuming, and expensive to add new features and capabilities to old software. This can hinder a company’s ability to respond to competitive pressures in the market. Because software technology advances at a more rapid rate than hardware, the gap between hardware functionality and software capability increases quickly.

For example, it is usually impossible to translate (or localize) old software for non-English-speaking markets, limiting the size of the company’s total market. Most older software was simply never designed to store or display non-English characters or fonts. The newest Microsoft and Linux platforms support multiple languages, so creating software for a global market is simply another built-in capability.

From a business and technical perspective, maintaining old software presents a management challenge as well. It can be difficult to recruit or to retain good engineers if they are to be tasked with working on old technology. Worse, as engineering turnover occurs, a company’s fundamental ability to maintain its product is eroded. As software developers with intimate knowledge of the inner workings of the system leave the company, it becomes more difficult for other software developers to make incremental upgrades or other changes or to even understand how the system works.

### **Rewriting the Software**

Is it possible to justify completely rewriting device software? Three years ago the answer was probably not. Most CEOs could not justify upgrading software simply to be on the newest platform. That always seemed like an expensive treadmill to be on, and it was

difficult to justify the expense relative to the business value.

Today, with the advent of new platforms and new tools, it may be that a company can’t afford not to upgrade. According to Microsoft expert Dean Samuels, “Today you have to look at the value of information in a networked world. The current Microsoft platform allows clinical data and images to be shared and acted upon in ways that simply weren’t possible just a few years ago. Connectivity and collaboration, which are built into the platform, allow clinicians, researchers, and administrators to not only share data with each other, but also with other technology-enabled systems and devices. That’s just one example of how the platform (in this example Microsoft .NET and Microsoft Embedded XP or CE .NET) adds very substantial business and clinical value simply by building a product on that platform.”

Indeed, the medical device industry is just now beginning to address basic issues such as interoperability and plug-and-play models, which have existed for years in other industries. A white paper was written June 22, 2007, by Julian M. Goldman and Sue Whitehead, MD, director and program manager, respectively, of the Medical Device Plug-and-Play Interoperability Program sponsored by Massachusetts General Hospital and the Center for the Integration of Medicine & Innovative Technology. The paper notes the following: “Unlike the interconnected plug-and-play world of modern computers and consumer electronics, most medical devices are designed to operate independently and do not employ open networking standards for data communication or for device control.”

Another substantial business benefit to using the newest platforms is dramatically improved competitiveness. New features can be designed, developed, and tested far more quickly and less expensively than with older technology. This capability allows companies to be far more responsive to customer requests, adding new features and driving additional upgrade and maintenance revenue. The building-block approach to development enables product features to be added quickly, and testing time is minimized. Because

software components that make up each building block have already been thoroughly tested, the test efforts can focus on integration and feature (or feature correctness) testing. Once a software component has been certified through the company's quality assurance process, that component can be reused allowing software development time to be even further reduced.

In general, the advantage of using the new development platforms is that much of functionality and capabilities that previously had to be hand coded is now built into the system. Communications protocols and data transfer to and from other devices and systems are built into the platforms.

The risk with such easy-to-use tools is that inexperienced programmers can quickly introduce errors, so expert project management and rigorous quality assurance procedures become even more critical.

### **Business Issues Drive Adoption**

One issue that will most certainly drive adoption is business competition. Over the next one to two years, as new venture capital-backed medical device companies come to market, they will not be constrained by legacy software. In fact, they will design their embedded and application-level software using the newest platforms. As a result, they will come to market with both techni-

cal and marketing advantages.

They will be more price-competitive because they will have spent far less time and money developing their software. Their devices will be easier to install and easier to learn and operate. Their application-level software will be of much higher quality and thus will generate fewer technical support calls and questions, saving them money on training and support.

Most significantly, with powerful new user-interface capabilities, their medical devices will be able to display and report clinical data and test results in more useful and meaningful ways. Similarly, with equally powerful embedded software, their devices will have the ability to gather, manipulate, analyze, process, compare, and contrast clinical data faster and in ways that will be more valuable for the medical professional.

Medical device companies face a technological crossroad. Those who choose to stay with their older platforms may find themselves losing significant market share or far worse.

### **Conclusion**

It is the software components of medical devices that will ultimately provide the most clinical and business value to end-users. More-powerful software, analyzing, measuring, and manipulating more-precise

and voluminous data from a series of seamlessly networked devices, benefits clinicians, healthcare providers, and researchers, but the key benefit accrues to patients.

After all, patients are our ultimate consumers. Although advances in hardware will continue to drive value for patients, it is the advances in software that have the potential to provide stunning breakthrough value.

When the medical device industry moves to the new, more-powerful commercial platforms, new emerging service models can be created for patients, clinicians, researchers, and care providers. Data that once lived in a stand-alone device can now be integrated with data from dozens of other systems. The ability to review and analyze data from many different perspectives allows clinicians to correlate information and see trends and data points that ultimately provide far more valuable services for patients.

Software tools now exist that allow medical device manufacturers to achieve such results faster and less expensively. The key will be for today's medical device manufacturers to break from their old ways of thinking, abandon their cultural inertia, and invest in the future of software technology. They can be certain that new competitors are already working with new software technology. ■

© Excerpted from MEDICAL DEVICE & DIAGNOSTIC INDUSTRY, January 2008. A Canon Communications LLC Publication.



**Full Spectrum Software is a highly specialized software engineering and software testing consulting firm that works exclusively with medical device and life sciences companies. The company's Quality Systems are based on FDA compliance guidelines. Full Spectrum offers a wide range of software engineering, design, development, V&V and testing services. Full Spectrum Software has been in business for 12 years and their clients include industry leaders such as Thermo Fisher Scientific, Toshiba Medical and Applied Biosystems as well as numerous smaller companies who are often working at the cutting edge of technology. The president of Full Spectrum Software, Andrew Dallas, is widely considered one of the leading experts in software for the medical device industry.**