

# OPTIMIZING TECHNOLOGY CHOICES FOR DEVICE SOFTWARE

How OEMs and developers can significantly reduce  
development costs and time-to-market

Jerry Krasner, Ph.D.

May 2005

**EMBEDDED MARKET FORECASTERS**  
**American Technology International, Inc.**

**Embedded Market Forecasters**  
Research and Consulting  
for Embedded Products,  
Markets and Channels



## I. Overview

A new economy doesn't replace the old economy – it transforms it. Such transformations are necessarily disruptive. They redefine the strategic interests of customers, reshuffle market segments, and reestablish market segment dominations. This study creates a clear migration path to the technologies that will redefine the embedded industry, as well as the embedded market segments that will benefit from enhanced connectivity, embedding security, and bandwidth availability.

The nuclear winter of the telecom market crash has long since past, and new and potential winners are staking out their strategies. RTOS vendors are moving to distance themselves from the commodity space that has long defined their marketplace to new and promising market segments that offer significant growth not only for themselves, but also for the OEMs and developers that use their products, tools, and services to gain significant growth in their markets.

As would be expected in such a competitive marketplace, OEMs and developers are confronted with claims, counterclaims, and distortions from RTOS and other software and tools vendors. Although EMF sees the competitive advantages and disadvantages of such ads, press releases, and sales promotions from the key RTOS vendors, we also recognize the confusion that they can cause to OEMs and developers that are seeking perspective on which markets are growing and which vendors can best serve their specific needs for a particular application.

For example, OEMs can certainly be confused by issues surrounding such requirements as POSIX conformance (and the difference between conformance and compliance), DO-178B level A, ARINC-653 conformance, and Common Criteria specifications, including Evaluation Assurance Levels (EAL) 5 through 7. Many OEMs and developers are unaware of the advantages of and need for FIPS 140-2 encryption modules, which are required for sales of communication and network connectivity software to the government.

With the embedded industry undergoing a broad expansion over many new market segments (including enterprise), these issues have broad and significant implications for future growth. In the medical arena, for instance, HIPAA (Health Information Protection and Accountability Act) regulations for patient privacy regarding patient processing, access to and storage of medical data, and financial transactions have precedents set in government and military security initiatives. Hence factual information regarding vendor claims and their importance to current, legacy, and future developments are essential to how OEMs and developers make purchasing and utilization decisions.

Embedded security is, we believe, the next great application and product differentiator for the embedded industry. Embedded security has become the focus of government, military, avionics, and telecommunications customers. DO-178B, EAL 7 (Common Criteria), ARINC653, POSIX 1003 conformance, and FIPS 140-2 not only have a major impact on expanding government purchases, but many of these performance standards are also being adopted by HIPAA and financial

application standards. The NSA (National Security Agency) has deemed that IPv6 will replace IPv4 (for security reasons, in addition to addressability) within two years. How do IPv6 and EAL impact FIPS 140-2, and how does this play into OEM decision-making?

We will be mentioning four RTOS vendors and their product offerings as examples in our presentation. These companies – Wind River Systems, Green Hills Software, LynuxWorks, and Accelerated Technologies (Mentor Graphics) – are significant companies that offer quality products and services. EMF looks at their product offerings and the standards (and verticals) that they support, and presents a perspective – based on extensive embedded design data – from which our targeted readers (OEMs, developers, and systems integrators) can make better decisions.

EMF conspicuously avoids any vendor or product recommendations, and attempts to make clear the facts surrounding such offerings and the time tables for standards conformance – and reasons expressed by the mentioned vendors – so that readers can draw conclusions based on their respective needs. EMF respects and holds in high esteem the vendors on whom we report, and we have selected them based on their record for quality in the embedded industry.

In this paper, EMF lays out its vision for growth in the embedded world and the role that the various technologies and standards play in helping OEMs and systems integrators expand their markets.

## II. Current and Growing Embedded Markets

### The Transition

The embedded marketplace is characterized by change – not the slow methodical change that characterized more commoditized markets, but rather a great upheaval. In prior years, the RTOS marketplace was relatively stable and on the verge of becoming commoditized. Then came the collapse of the telecom space and the battering of the telecom-heavy RTOS markets. But technology represents a resilient business opportunity space, and it wasn't long before the market segment transformed itself into a new optical- and wireless-based market with enormous potential. Enterprise- and government-based applications have begun to manifest. Embedded design criteria far surpass the "five-nines" reliability criteria of the old telecommunications requirements, and traditionally "embedded" vendors are beginning to find more expansive markets in enterprise and military markets – and these greatly exceed embedded markets of the past.

The embedded market is growing in one direction by expanding embedded vendor markets to include enterprise and military. Expansion in another direction addresses the development and running of device software in ways that cut both costs and time-to-market, and produce better software that's more reliable. The graphs later in this paper all point to the problems of creating software on time, on budget, and with the desired functionality. The problem exists today, and the solution is Device Software Optimization (DSO): the creation of device software better, faster, at a lower cost, and more reliably.

Driven by an insatiable demand for newer products and technologies, embedded design engineers are confronted not only with complex design considerations, but also with limited resources and shortened product design cycles.

Product opportunity windows are fleeting, and time-to-market issues dominate design considerations.

Those vendors with the vision to grow into these opportunities are reshaping the expanded embedded markets for the future.

The stability of certain design applications and the upheaval of other market segments place a considerable burden on embedded design engineers. They must respond to limited windows of opportunity, even as many of them are forced to address greater design complexities.

Driven by an insatiable demand for newer products and technologies, embedded design engineers are confronted not only with complex design considerations, but also with limited resources and shortened product design cycles. Faced with choosing among a panoply of microprocessors and operating systems (which might also require servicing legacy designs), they find that the distinction between hardware and software development tools is rapidly vanishing, as issues of hardware-software integration and hardware-software partitioning become of greater concern. The growth of co-development and co-verification tools, as well as the growth of simulation-modeling tools with automatic code generation within the embedded marketplace, reflects these changes.

Beyond the details of the direct and indirect market forces that create an environment where competitive edges frequently determine success and failure, lies an industry of immensely talented and diverse individuals whose skills and insights add opportunity and uncertainty within the competitive landscape. Technological insight and ability are not scarce commodities in today's era of high technology. Current technologies are under assault by a wave of newer computer architectures, operating systems, and I/O configurations. Product opportunity windows are fleeting, and time-to-market issues dominate design considerations. What invariably makes the difference between success and failure is the knowledge that comes with accurate insight into these market-driving external forces, as well as insight into the concerns, desires, and thought processes of customers who decide to purchase a particular product or deal with a particular vendor.

The vendors who will emerge to dominate the expanded embedded space will be those who offer developers broad use of embedded tools and OSes, in addition to their own proprietary commercial offerings. These vendors, by virtue of having met the stringent requirements of government/military standards and performance assurances (e.g., POSIX conformance, Common Criteria, FIPS 140-2, etc.), will be in an advantageous position to address the stringent needs of enterprise, medical, and financial market requirements.

The medical privacy and confidentiality standards set forth by HIPAA and the U.S. Congress legislative financial transaction GLB (Gramm-Leach-Bliley) Act requirements are most likely to adopt the standards set forth by the NSA. It is very unlikely that different standards would be invoked by these governing groups than those set forth by the NSA or government-tested and validated Evaluation Assurance Levels. This means that vendors, developers, and OEMs that conform to the more rigid military and government standards will find themselves in a more favorable position to address medical and financial application markets than those that can't meet such standards.

## One Size Doesn't Fit All

In prior years, embedded RTOS vendors attempted to promote their RTOS/IDE and tools as a one-stop shop for all applications and all developers. Wind River – the market leader – developed specialized versions of VxWorks for a multitude of different verticals. But they have come to realize that VxWorks may not be the appropriate OS for all applications, and they have concluded that, in order to better serve their current and prospective customers, they need to create an environment wherein developers can work with a variety of tools and operating systems from different vendors within their Workbench IDE.

Green Hills Software has their MULTI IDE, which is similarly capable; and Mentor's Accelerated Technology has released EDGE to allow customers to work with Linux and Nucleus within the same development environment. LynuxWorks has fashioned their Linux offering to work within their long-established LynxOS real-time OS.

Wind River has redefined their expanded market plans under the name of "Device Software Optimization" and has moved to enable developers (real-time and non-real-time) to develop with Linux, as well as be able to use Eclipse-based tools and offerings. The common theme between the aforementioned vendors is that they enable and support developers to use Linux for non-real-time and non-mission-critical applications within their own IDE.

It is interesting that the real-time OS vendors, rather than the dedicated embedded Linux vendors, have provided the greatest tools support for Linux developers and have expanded the developer's design and integration options.

The embedded Linux marketplace has been hampered by the dearth of design tools available to embedded developers. The business model makes it difficult for such vendors to differentiate their offerings, since they all address the same specification. Only the commercially viable vendors with expanded offerings can afford to provide integrated support, thereby expanding and enhancing the development process for open-source developers.

Table 1 illustrates the diversity of embedded applications and the software requirements for each category. At issue is how an embedded OS vendor can provide an OS and tool sets that support a developer or OEM that may have multiple requirements for operating systems and applications.

The real-time OS vendors, rather than the dedicated embedded Linux vendors, have provided the greatest tools support for Linux developers and have expanded the developer's design and integration options.

Table 1: Device Requirements

Device Category/Vertical	Representative Devices	Software Requirement
Avionics	guidance, controls, communications	Mission-critical, real-time, highly secure
Automotive/Transportation	infotainment, ABS, drivetrain, GPS	mostly non-real-time; highly secure
Consumer Electronics	audio/video digital media, games	non-real-time
Gateway Server	routers, gateways, modems, appliances	some real-time, secure
Industrial Controls	test and measurement, controls, instrumentation	some real-time, secure
Office Automation	printers, projectors, communications	non-real-time
Internet Appliance	Web pads, browser phones, access devices	non-real-time, some security
POS/kiosk	POS terminals, modems, gaming	non-real-time, secure
Datacom	servers, routers, DAQ	non-real-time, secure
Telecom	SS7, wireless, optical	non-real-time, secure
Set-Top Box	satellite, cable, DSL, modems	non-real-time
Military	ships, integrated communications, weapons	mission-critical, real-time, highly secure

This is the challenge to vendors: to not only increase their market share, but also position themselves in a broader and more expansive marketplace. As we will see, each vendor has taken a different approach to this opportunity.

**Design Completions and Proximity to Pre-Design Expectations**

Embedded designs are notorious for experiencing delays and cancellations. This problem is expected to become worse as design complexity increases, reliability requirements are made more stringent, and security concerns dominate connectivity issues.

Tables 2 and 3 illustrate the divergence between early and late design completions. This data has been remarkably consistent over the past four years, changing by only a few percentage points. It is also interesting to note the proliferation of 32-bit, DSP, and FPGA designs and the corresponding design experience.

Embedded designs are notorious for experiencing delays and cancellations. This problem is expected to become worse as design complexity increases, reliability requirements are made more stringent, and security concerns dominate connectivity issues.

It is not surprising that medical, military, consumer electronics, and industrial automation verticals fared better than most – but there was little to get excited about.

Table 2: Design Completions According to Schedule

Design	Ahead	Behind	Cancelled	Outsourced
8-bit	17.1%	53.7%	13.7%	11.3%
16-bit	18.0%	49.5%	12.6%	12.5%
32-bit	15.4%	53.5%	13.1%	11.4%
64-bit	17.7%	47.3%	11.9%	16.8%
DSP	17.2%	54.8%	12.9%	11.3%
FPGA	15.8%	54.3%	12.6%	11.0%

As we will report later, some of the problems encountered in design delays and design cancellations were associated with the number of methodologies employed and the availability of a higher tools budget (which permitted the developer to purchase more and different design tools).

It is anticipated that the newer and more comprehensive IDEs being brought to market (e.g., Wind River’s Workbench and Green Hills’ MULTI) will provide a common design base that will enable developers to work across a number of OSes and processors with a single design environment.

With design complexities, reliability standards, and security requirements increasing, a newer approach to embedded design is clearly needed.

Table 3: Design Completions by Vertical Market

Vertical Market	Ahead	Behind	Cancelled	Outsourced
Auto-Transport	14.8%	55.4%	15.1%	13.1%
Avionics	15.1%	52.0%	11.8%	15.2%
Bus Mach & Peripherals	15.1%	52.9%	14.8%	11.6%
Consumer Electronics	17.2%	52.3%	14.8%	11.9%
Datacom	11.8%	57.2%	16.5%	13.2%
Telecom	10.6%	60.2%	18.3%	7.5%
Electronic Instrumentation	18.3%	57.3%	13.3%	11.1%
Industrial Automation	19.1%	51.3%	13.0%	8.1%
Medical	18.1%	56.2%	11.6%	11.3%
Military	17.6%	52.1%	8.3%	14.3%

Better design results can be expected from a more comprehensive and useful IDE for a number of reasons, including:

- Developers can become skilled in one or two processes, but it is difficult to be fully proficient in more
- Developers have fewer training classes to attend and fewer user manuals to consult, resulting in less downtime
- Development teams with fewer processes have a common language based on those processes
- Developers who move from one project to another can more quickly participate when the new project's processes resemble the prior project's processes
- Having fewer processes to choose from increases the chance a developer can transition smoothly
- Management can focus on product quality and staff quality, rather than on process management and churn

EMF data shows that an increasing number of embedded designs interface directly, indirectly, or both with an enterprise system. Table 4 illustrates the percentage of such designs across 10 vertical markets.

An examination of Tables 2 through 4 strongly suggests that a better design platform is necessary to address the increased marketplace that enterprise (and military) is creating. The broader embedded marketplace needs to address these issues. This shift in the embedded market is known as Device Software Optimization, or DSO.

Table 4: Enterprise Interface with Embedded Designs

<b>Does Your Embedded Design Interface with an Enterprise System?</b>					
	<i>(base = 642 respondents)</i>				
<b>Interface</b>	<b>Any</b>	<b>Internal</b>	<b>External</b>	<b>Both</b>	<b>None</b>
Auto-Transport	62.1%	20.7%	10.3%	31.0%	37.9%
Avionics	40.9%	15.9%	9.1%	15.9%	59.1%
Bus Mach & Peripherals	78.6%	32.1%	17.9%	28.6%	21.4%
Consumer Electronics	54.3%	8.6%	14.3%	31.4%	45.7%
Datacom	77.8%	15.6%	17.8%	44.4%	22.2%
Telecom	75.4%	16.9%	16.9%	41.5%	24.6%
Electronic Instrumentation	62.9%	25.7%	5.7%	31.4%	37.1%
Industrial Automation	54.9%	26.8%	6.1%	22.0%	45.1%
Medical	59.5%	29.7%	10.8%	18.9%	40.5%
Military	47.9%	17.6%	7.7%	22.5%	52.1%

The real impact of design complications can be seen in Table 5. Developers were asked to state how close their final designs came to their pre-design expectations. They were given choices of within 10%, within 20% (meaning between 11% and 20% of expectation), and so forth. The question was asked separately for performance, systems functionality, features, and schedule.

Developers reported that their final designs were not within 50% of expectation an astounding 26% of the time (in 1999) to 33% of the time (in 2003) in ever-increasing levels. These data are not shown in Figure 5, but have been published in prior EMF reports. In 2004 and 2005, that figure dropped to around 3%, although it rippled down to lesser but not satisfactory levels.

Table 5: 2004 Design Results

How Closely Does Your Final Design Approximate Your Pre-Design Expectation?						
	Within 10%	Within 20%	Within 30%	Within 40%	Within 50%	Not Within 50%
Performance	38.7%	26.7%	14.1%	2.1%	15.7%	2.6%
Systems Functionality	45.0%	25.1%	6.8%	3.1%	17.3%	2.6%
Features & Scheduling	28.0%	26.4%	18.7%	7.3%	15.0%	4.7%

The results presented in Table 5 are the best that have been recorded in over six years of surveying – yet 48% of final designs are not within 20% of pre-design expectation (for features and scheduling, for example), which remains a questionable design result. With design complexities, reliability standards, and security requirements increasing, a newer approach to embedded design is clearly needed.

### III. Standards and Requirements Driving the New Embedded Marketplace

As the embedded software and tools marketplace seeks a path to new and profitable markets, the strategies of the past are being cast aside. Vendors now seek to use standardization to their advantage, rather than pursue proprietary, incompatible solutions.

Developers and OEMs can benefit from the fact that the use of software standards can reduce costs and enhance the design process:

- By increasing code reusability, project to project, system to system
- By reducing the learning curve for developers
- By enabling the use of open-source or third-party code
- By enabling swift redeployment of developers, project to project
- By enabling the use of third-party products that are integrated and tested to work in that environment (e.g., Eclipse)

The embedded software and tools marketplace have become largely commoditized. RTOS and IDE vendors are competing in a virtual zero-sum game. What the embedded industry needed was new and expanding markets – particularly those that dwarf the established ones. New and nontraditional markets are opening and expanding rapidly, creating new opportunities for smaller vendors as well.

One way that the embedded market is expanding is by moving into new or larger markets, such as the enterprise and the military. But another way to expand the embedded marketplace is by addressing the problems of creating device software, not just where to place that software.

No matter what market a company sells device software into, that company is going to face problems developing it on time and on budget, with the right features, high quality, and reliability. That's where DSO comes in.

Tables 2 to 5 clearly show that no matter what market a company sells device software into, that company is going to face problems developing it on time and on budget, with the right features, high quality, and reliability. That's where DSO comes in.

Niche market opportunities are characteristic of expanding markets, as engineering-depleted larger companies and OEMs need smaller and faster-responding partners in order to deliver superior products on time and within allowable cost structures.

Three items distinguish the new device software optimization marketplace:

- Interoperability
- Guaranteed software quality
- Enhanced security

By providing any or all of these, any vendor can significantly broaden their market opportunities.

Let's examine each of these characteristics.

## Interoperability

As software complexity increases and applications are used more widely, it is essential to be able to interoperate software designs, many of which have been developed using different RTOSes. This need is also amplified by the requirement to be forward- and backward-compatible in order to upgrade systems without major software rewriting, and to reuse existing applications in newer upgrades.

Looking at the U.S. Navy, for example, consider the findings of a conference sponsored by the Carnegie-Mellon Software Engineering Institute.

The next generation of Navy ships will have a 30- to 50-year life expectancy. The Navy discovered that the cost of on-board personnel (sailors) is extremely high, and the expense forecast for current staffing levels would double the cost of the ship itself. With an all-volunteer military, there is also the uncertainty of maintaining staffing levels. Hence, there is a major incentive to reduce the number of sailors required on ships by automating processes using highly reliable and secure computing.

Central to this strategy is the ability to create application software that can be used and upgraded across a large number of installations. The mission can be summarized as follows:

- Determine how small a crew is needed to effectively operate all systems
- Determine how uniformity can be implemented across all ships and systems, enabling a greater uniformity and expedience in training
- Software interfaces must be uniform across all operating systems (e.g., POSIX)
- As much as possible, existing software should be reusable for newer and upgraded systems
- Rewriting of software should be held to a minimum

Under the Program Management Office (PMO 450) of NAVSEA, there is a move toward electric-drive ship mechanisms, whereby:

- Helm control can be generic
- Maneuver control requires advanced software
- Uniformity, redundancy, and common GUIs are required
- It is essential that electric drive mechanisms can be moved from ship to ship
- Extensive software development and lots of common, transferable computing power are needed

## Guaranteed Software Quality

Aerospace and Defense manufacturers must optimize their device software development to provide next-generation applications with more interoperability, safety, security, and connectivity. Central to this need is the ability to qualify both the performance of developed software and the developmental process that created the software in the first place. This “best practice” assurance is central to defense acquisitions and avionics programs across the board.

Looking at U.S. defense implementations: For coalition and allied forces to counter current and future threats successfully, they must operate worldwide with speed, agility, and flexibility. They must have access to accurate, current, and timely information, and the capability to share this information securely. In order to meet these requirements, aerospace and defense manufacturers are faced with:

- Integrating devices to the Department of Defense’s (DoD) Global Information Grid, a key element of future combat power. To support the continued growth and advancement of net-centric warfare capabilities, the DoD requires that all devices will be IPv6-compliant by 2008 (the NSA has accelerated the requirement that all devices will be compliant).
- A demand for mission-critical systems, with secure connections, which meet Evaluation Assurance Levels (EAL1-7) and DO-178B safety certification levels (A-E).
- Adhering to open standards like POSIX and ARINC653 to facilitate device interoperability

Aerospace and Defense manufacturers have been challenged to optimize their device software development to provide next-generation applications with more interoperability, safety, security, and connectivity. If you are the Joint Commission of Accreditation of Health Organization (JCAHO) and are ultimately responsible for setting software and patient information security requirements (as will be required under HIPAA), it seems like a no-brainer to look at the security and software performance requirements set forth by the government for defense and avionics applications.

A similar case can be made for financial data transfer and acquisition applications (now governed by GLB legislation). And enterprise applications that involve company confidential information, such as CRM, inventory, pricing schedules, and customer data files, face similar concerns that may be addressed through application of the government’s solutions for defense and avionics requirements.

It seems like a no-brainer to look at the security and software performance requirements set forth by the government for defense and avionics applications.

## Enhanced Embedded Security

Embedded systems are responsible for the availability and functionality of many critical systems, from factory automation to gas pipeline monitors to networking equipment. Unfortunately, the critical importance of embedded systems is seldom matched with a strong, comprehensive security infrastructure. Some of the critical security issues presented by modern embedded systems are:

- Diverse network-connected embedded systems use combinations of custom and COTS software, the details of which are typically known only to the vendor of each embedded device, making vulnerability assessment, risk analysis, and patch management difficult
- Many embedded protocol implementations derive from older versions of open-source software like OpenSSL and the BSD TCP/IP stack, resulting in vulnerabilities to known attacks, which have since been patched in the main software distributions
- Many other protocol implementations are built entirely from scratch, and have not benefited from years of public analysis and repeated attacks, resulting in unproven protocol implementations that may be vulnerable to attack
- Even when vulnerabilities are identified, patches must be developed for each device or device family by the vendor, requiring tight collaboration between embedded software developers and the OEMs that build devices based on the developers' software
- Deployment of software patches is even more difficult, expensive, and time-consuming than the most elaborate mobile/remote patch management systems for PCs and PDAs, making the total cost of a vulnerability in an embedded system much higher, and thus the motivation to patch that vulnerability much lower
- Most network-aware embedded devices lack sufficient management and auditing functionality, making centralized configuration and monitoring difficult and costly, and severely limiting the data available for attack pattern detection and after-attack forensic analysis
- Embedded systems are not always considered an IT responsibility, and thus often fall outside IT control, resulting in lax policy enforcement, minimal configuration management and auditing, distorted risk analyses, and little or no integration with enterprise security tools

Remediation of these issues requires a concerted effort on the part of commercial and custom embedded software developers, OEMs building embedded systems, vendors selling them, and customers purchasing and implementing products based on network-aware embedded software. Until information security becomes a strategic technology for embedded systems developers, their products will continue to be characterized by complacency and vulnerability.

However, embedded vendors and OEMs don't have time to be complacent regarding security. The U.S. government has mandated that all equipment (including embedded designs) that provides connectivity for any level/type of connectivity must utilize FIPS 140-2-certified modules for all purchases by the U.S. government, as must all procurements by OEMs that have government contracts. The NSA has decreed that there are no exceptions to this requirement. Device software developers could well increase the reliability of their software by complying with this and other security standards.

## IV. Case Study: U.S. Defense Priorities and Best Practices

The following information is presented with permission of Steven Roemerman, CEO of Lone Star Aerospace, a market research organization that specializes in high-level analysis of major defense-related companies and senior government officials. Using the extensive EMF database and Executive Dashboard, Lone Star compared the results of their respondents (very high-level decision-makers) with the detailed user data available from EMF to search out best practices.

The following materials are the conclusions of Lone Star and are presented herein as they present a potential roadmap for embedded vendors and OEMs to follow emerging opportunities.

Lone Star conducts annual surveys of high-ranking executives and government program managers. Industry respondents are usually at the VP level or just above, while government program managers are usually at the rank of general/admiral or the civilian equivalent.

Lone Star asked a number of questions about how the DoD, DHS, and industry are doing – and inquired as to whether we as a nation are focused on the right things.

### Looking at Acquisition, Contracting, and Execution

#### **“What must we do better to provide war fighters with what they need?”**

Areas indicated as needing the most attention:

- Shorter procurement cycle time
- Contractor technical talent pool
- Improved development of software in weapon systems
- Improved Program Stability
- Use of innovative small and medium-sized businesses by prime contractors
- Development of DoD IT systems (logistics, mission planning, etc.)

Claimed management and development practice problems:

- We don't have enough U.S. citizens who can create software products and who have a strong background in other disciplines (mathematics, engineering, control theory, etc.)
- Software as a discipline has shattered into a number of sub-specialties (enterprise, real-time control, DSP programming, etc.), and managers often lack the ability to find or distinguish the appropriate talent, methodology, and tools for product development
- We don't have enough of the best minds who want to pursue defense technology – the Apollo generation is retiring, and the best new graduates don't see defense as cutting-edge
- We don't know if our processes, tools, certifications, and methodologies are working
- We participate in so-called best practices because we are expected to, not because we know they work

- Government-sponsored, best-practice clearinghouses include a seemingly random mix of practices that work and those that don't
- Proponents of some claimed best practices don't actually deploy them

These results are consistent with problems experienced by developers of traditional embedded system – and they point to an opportunity. Design cycles are becoming more complex, and managers don't know what "best practices" are, since there had not previously been any definitive data indicating which practices produce better software design results.

## **Looking at Defense Usage of Software and Best Practices Analysis: The EMF Database**

Lone Star used the EMF 2004 Database to examine software usage and methodologies, from which it discerned "best practices" utilization for defense applications as compared with other usages of embedded software. The database used by Lone Star for this analysis is the same that has been available to EMF subscribers.

### **Lone Star's conclusions are as follows.**

Defense utilization compared with other vertical embedded markets:

- Defense respondents have slightly more development methodologies available to them
- Defense respondents are about 50% more likely to use simulation and modeling in their developments
- Defense respondents are somewhat more likely to use object-oriented development methodologies
- Defense Projects with embedded software are 2.75 times more likely to finish late than to finish early
- Defense firms with high software concentrations are more likely to have staffing shortages
- Defense systems are becoming increasingly dependent on embedded software
- Defense projects are not worse, on average, than industry in general; but they fall short of best in class (which seems to be medical electronic projects)

### **Loss of functionality in embedded designs was the major predictor of schedule slippage.**

Loss of functionality (LOF) is the percentage of expected system functionality which was actually achieved in the final product. Lone Star found that loss of functionality was an indicator of development integrity:

- LOF was the most consistent measure tested (that is, loss of functionality was the most consistent indicator of development integrity)
- LOF is also critical in every industry vertical that EMF surveys, since missing functionality can have significant regulatory or competitive implications

## Looking to Medical Developers as a Defense Comparison

With a measure that seemed reliable (LOF), Lone Star wanted to find a subset of EMF respondents with better results than DoD. Medical developers reported more contrast than other groups they tested.

Medical products have several similarities to DoD products, in that they:

- Are highly regulated
- Have unique specifications and standards
- Are procured by formal methods
- Involve life or death consequences
- Are increasingly dependent on software for advanced signal processing management, sensor management, networked force management, etc.

## So, what can defense developers learn from the success of medical developers?

Project results varied between defense and medical:

- Defense respondents reported that 17% of their projects were completed ahead of schedule (about the average across the entire survey)
- But medical respondents reported that **almost 39%** of their projects were completed early, meaning performance was **2.75 times better** than that of defense
- The average defense project lost 50% more functionality than the average medical project
- 8% of defense projects finished late, the same as the survey average
- But medical respondents reported that **less than 40%** of their projects finished late

## Fewer methodologies/processes are better.

Lone Star found from the EMF data that this was, in fact, the case. They drew their conclusion from results indicating the following advantages are gained by reducing methodologies and processes:

- **More mastery:** Developers can become skilled in one or two processes, but it is difficult to be fully proficient in more
- **Less downtime:** Developers have fewer training classes to attend and fewer user manuals to consult
- **More coherent vocabulary:** Development teams with fewer processes have a common language based on those processes
- **More skills portability:** Developers who move from one project to another can participate more quickly when the new project's processes resemble the prior project; having fewer processes to choose from increases the chance a developer can transition smoothly
- **Better focus:** Management focuses on product quality and staff quality, not on process management and churn

## V. FROM DEVELOPER NIGHTMARES TO VENDOR SOLUTIONS: STANDARDIZING ON CHOICE

### Conclusions

There are certain unassailable facts that confront embedded developers on a daily basis:

- Designs are becoming increasingly complex
- Market windows of opportunities are shrinking
- "One size fits all" technology no longer works
- Device software has to be created faster, better, more cheaply, and more reliably
- Device manufacturers need to optimize their technology choices for software and development environments to ensure that their technical and business goals are met

EMF and Lone Star data have clearly shown that design results are greatly enhanced by minimizing the number of methodologies that must be managed, and by reducing the number of tool choices that developers select from and use.

Evidently, a developer or OEM will benefit from selecting a development environment that:

- Meets stringent standards for software reliability, failure protection and security
- Provides the greatest flexibility for selecting the OS most appropriate to the application (real-time or non-real-time, based on power requirements, footprint, etc.), the processor of choice, and the tools most appropriate to the design requirements
- Enables the use of simulation modeling, rapid prototyping, and automatic code generation

### How Embedded OS and Tools Vendors Are Responding to the Challenge

#### Wind River

Wind River Systems has been a leader in refocusing their efforts to expand the current embedded industry. With the introduction of Device Software Optimization (DSO), the message is clear to enterprise developers, and it is shared by other RTOS vendors: Embedded design requirements far outweigh the five-nines requirements of enterprise design. Why use outdated and inefficient design techniques, when better and more affordable solutions are available?

Wind River's platforms tightly integrate a rich set of operating systems, development environments, and middleware with services to provide a complete foundation that meets the specific requirements of a vertical market. Wind River claims that their customers can now reduce effort, cost, and risk, and optimize quality and reliability at all phases of the device software development process, from concept to deployed product. This makes Wind River the market leader in DSO.

Wind River's Workbench is an example of creating high-level design capabilities in an expansive and easy-to-use development environment.

Why use outdated and inefficient design techniques, when better and more affordable solutions are available?

Workbench provides comprehensive tools for code analysis and debugging, using an extensible framework based on the open-source Eclipse standard. It includes a broad range of features, such as: a high-performance on-chip debugging solution; target operating system monitoring; runtime software component configuration; bit-level, assembly-level, source-level, and operating system-level visibility and control; source code analysis; and software behavioral analysis tools. In addition, Workbench offers multiple host environment support and is designed to be target-operating-system agnostic, supporting VxWorks, Linux, and others to be announced. It also supports multiple target connections and provides multiple process, task, and thread debugging support.

Workbench enables developers to increase collaboration, reuse, and flexibility within their projects; eliminates “learning curve” productivity losses and builds portable expertise; enhances self-support capability; improves ROI for acquisition, training, and extensions; offers a development suite for the entire development life cycle, from hardware bring-up to product test and manufacture; and offers seamless integration with third-party and in-house tools for customization and scalability, using the extensible Eclipse framework.

## **Accelerated Technology**

Accelerated Technology (ATI – Mentor Graphics) has redirected their efforts to a common design platform that they call EDGE.

Accelerated Technology based their last generation of tools on Microsoft Visual Studio, which offered the only available standard at the time. But it was proving difficult to meet the emerging group of embedded developers who wanted a true cross platform (including Linux host support) and a multi-core embedded development environment. They noticed that Eclipse was emerging as a standard in the enterprise world, was a good technical solution, and had a business model that allowed them to offer a complete solution to embedded developers. Nucleus EDGE (Embedded Developers Graphical Environment) utilizes the latest Eclipse platform (v3) and realizes all of ATI’s embedded technology as Eclipse plug-ins.

ATI developed a number of true embedded tools that all interoperate under the Eclipse framework and provide a complete development suite for embedded developers who are using Nucleus, their own proprietary RTOS, or no RTOS. Cross-compilers are a very target specific part of the embedded software tool chain, and although code is generally written in ANSI C, ATI believes that the compiler is the last part of the tool chain that engineers are willing to change. Within Nucleus EDGE, they offer a number of compiler options, allowing users to select their favorite, but still get all the benefits of Eclipse.

## **Green Hills Software**

Green Hills Software has developed MULTI. The MULTI Integrated Development Environment (IDE) provides a complete set of tools for the optimization of embedded software reliability, performance, code size, and time-to-market. Designed specifically to address the requirements

of embedded software developers, the MULTI IDE has been enhanced over more than 13 years, based on input from thousands of users and Green Hills Software's own use of MULTI for the development of all of its products, including MULTI itself.

The MULTI IDE is marketed to organizations that want to deploy a common development environment across multiple diverse projects. It supports development in C, C++, and FORTRAN; the AdaMULTI IDE adds support for Ada. It is available for all major 32- and 64-bit processor architectures, embedded and general-purpose operating systems (including velOSity, INTEGRITY, VxWorks, Linux, and in-house), and development hosts. In addition, the MULTI IDE is integrated with other popular development tools, including design tools, configuration management tools, source code editors, and compilers.

Linux is fully supported by MULTI, with the same capabilities as their other OS integrations:

- As a development host for most of the target processors and operating systems that they support
- As a target operating system (from Linux, Windows, Solaris, and HP-UX hosts)
- For native (self-hosted) x86 development

Green Hills has made no secret of their strong feelings that developers should use their Integrity OS, rather than Linux, and has further publicly expressed their view that MULTI is far superior to Eclipse. Green Hills states that if a developer does want to use MULTI with Eclipse (which is likely, because they are using an Eclipse-based tool), the MULTI Debugger and Green Hills C/C++ Compilers can be launched from the standard Eclipse C/C++ Development Tools (CDT) environment.

An interesting aside is that while Green Hills tries to discourage developers from using Linux as an OS and Eclipse as a development environment, Wind River and ATI conversely encourage developers to use their preferential OS and tools with their IDE, as well as within the Eclipse environment.

## **LynuxWorks**

LynuxWorks has integrated their LynxOS real-time operating system with their BlueCat Linux OS to provide real-time interoperability for mission-critical and Linux design options. LynuxWorks' Luminosity Eclipse-based Project Management supports LynuxWorks C/C++ projects for users who want to directly edit/manage their makefiles. LynuxWorks' Managed Make C/C++ Project support is also provided for users who want Luminosity to manage makefiles automatically. Developers can use the Luminosity Kernel Wizard to create C projects that build the LynxOS kernel or BlueCat Linux kernel.

### About the Author

Jerry Krasner, Ph.D., MBA is Vice President of Embedded Market Forecasters and its parent company, American Technology International. A recognized authority with over 30 years' embedded industry experience, Dr. Krasner was formerly Chairman of Biomedical Engineering at Boston University, and Chairman of Electrical and Computer Engineering at Wentworth Institute of Technology and Bunker Hill Community College. In addition to his academic appointments, Dr. Krasner served as President of Biocybernetics, Inc. and CLINCO, Inc., Executive Vice President of Plasmedics, Inc. and Clinical Development Corporation, and Director of Medical Sciences for the Carnegie-Mellon Institute of Research. Earlier, he was Senior Engineer at the MIT Instrumentation Laboratory.



Dr. Krasner earned BSEE and MSEE degrees from Washington University, a Ph.D. in Medical Physiology / Biophysics from Boston University and an MBA from Nichols College. He is a visiting professor at the Universidad de Las Palmas (Spain) where he is recognized for his work in neurosciences and computer technology.

### About Embedded Market Forecasters

The premier market intelligence and advisory firm in the embedded technology industry, Embedded Market Forecasters (EMF) is the embedded market research division of American Technology International, Inc. We specialize in providing high-quality data and expert analysis to support our clients' ability to assess the opportunities, risks, and competitive issues involved with developing and deploying embedded technologies. EMF has extensive experience providing both multi-client and custom research on topics including embedded boards, buses, software, hardware and development tools markets as well as embedded technology applications including embedded systems, digital signal processors (DSPs), FPGAs, single board computers, communications/IT, and multimedia. Our clients range from startups to Global 100 companies worldwide. Founded by Dr. Jerry Krasner, a recognized authority on embedded markets, product development and channel distribution, EMF is headquartered in Framingham, Mass.

## Glossary

<b>ABS</b>	Anti-lock braking system for automobiles
<b>ARINC-653</b>	Aeronautical Radio, Incorporated: a nonprofit corporation owned by member airlines to define form, fit, and function of avionics equipment. Spec for hardware partitioned APEX (APlication/ EXecutive) interface for integrated modular avionics systems and other safety-critical environments
<b>BSD</b>	The version of UNIX® developed at the University of California, Berkeley
<b>Common Criteria</b>	A systems-wide security capability defined under ISO-15408
<b>COTS</b>	Commercial-off-the-Shelf: represents “standard” products
<b>CRM</b>	Customer Relationship Management system
<b>DAQ</b>	Data Acquisition system
<b>DO 178B</b>	An assurance level that provides a level of confidence to the safety goals of software designs
<b>DoD</b>	U.S. Department of Defense
<b>DHS</b>	U.S. Department of Homeland Security
<b>EAL</b>	Evaluation Assurance Level, part of the Common Criteria
<b>Eclipse</b>	Organization promoting an open-source software development framework
<b>EMF</b>	Embedded Market Forecasters: <a href="http://www.embeddedforecast.com">www.embeddedforecast.com</a>
<b>FIPS 140-2</b>	NSA standard defining the use of an encryption module for network and other communications
<b>GLB</b>	Gramm-Leach-Bliley legislation (1999) defining security for financial transactions
<b>GPS</b>	Global Positioning System

<b>GUI</b>	Graphical User Interface
<b>HIPAA</b>	Health Information Protection and Accountability Act
<b>IDE</b>	Integrated Development Environment for developing software
<b>I/O</b>	Input/Output system
<b>IT</b>	Information Technology
<b>JCAHO</b>	Joint Commission of Accreditation of Health Organization
<b>LOF</b>	Loss of Functionality
<b>NAVSEA</b>	Naval Sea Systems Command
<b>NSA</b>	National Security Agency
<b>OEM</b>	Original Equipment Manufacturer
<b>OS</b>	Operating System
<b>PDA</b>	Personal Digital Assistant
<b>POS (POS/kiosk)</b>	Point-of-Sale system
<b>POSIX</b>	Portable Operating System Interface Standard: a specification for achieving application portability
<b>RTOS</b>	Real-Time Operating System
<b>SS7</b>	Common Channel Signaling System 7: global standard for telecommunications
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol: common protocols that govern how data are moved in a network