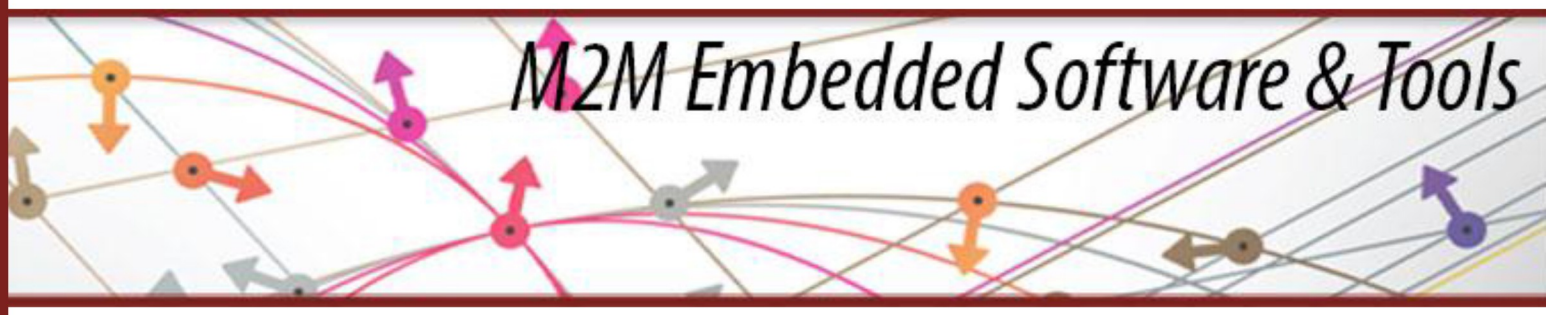
MARCH 2014

WHITE PAPER

THE FOLLOWING IS AN EXECUTIVE WHITE PAPER ON:



By: Chris Rommel – *Vice President*

Brewing Embedded Market Success with Java

Exclusive License to Distribute:
Oracle

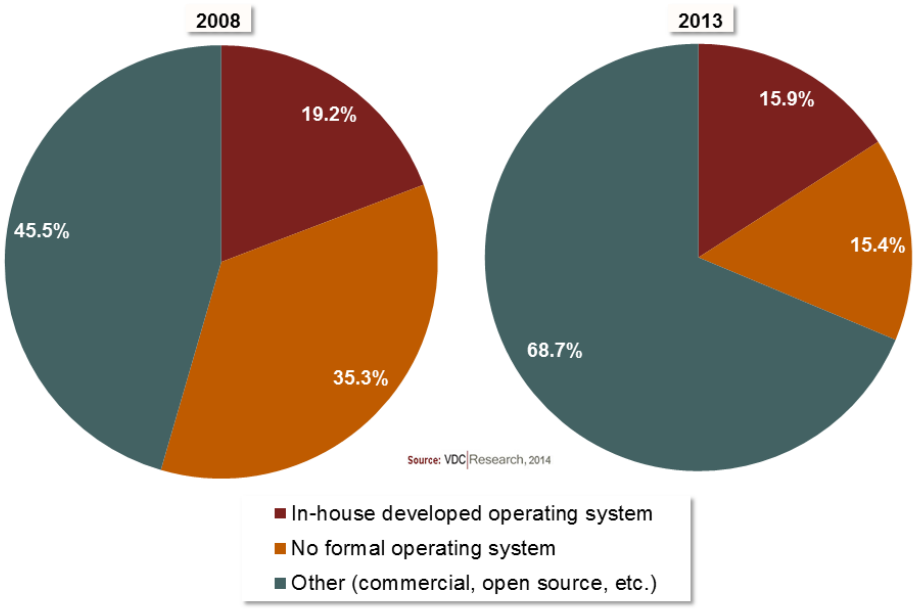
VDC Research
Insights for the Connected World

VDC Research
Insights for the Connected World

Embedded systems have infiltrated nearly every facet of our lives. Expectations for omnipresent connectivity and real-time, contextual computation are changing the face of embedded devices and the technologies that engineering organizations use to bring those products to market. These increasingly intelligent and connected devices are now forming the foundation of the Internet of Things (IoT). The conveniences and use cases borne from this IoT span a wide range of device and application classes, from the connected car and mobile point-of-sale devices to smart meters and wireless patient monitoring systems.

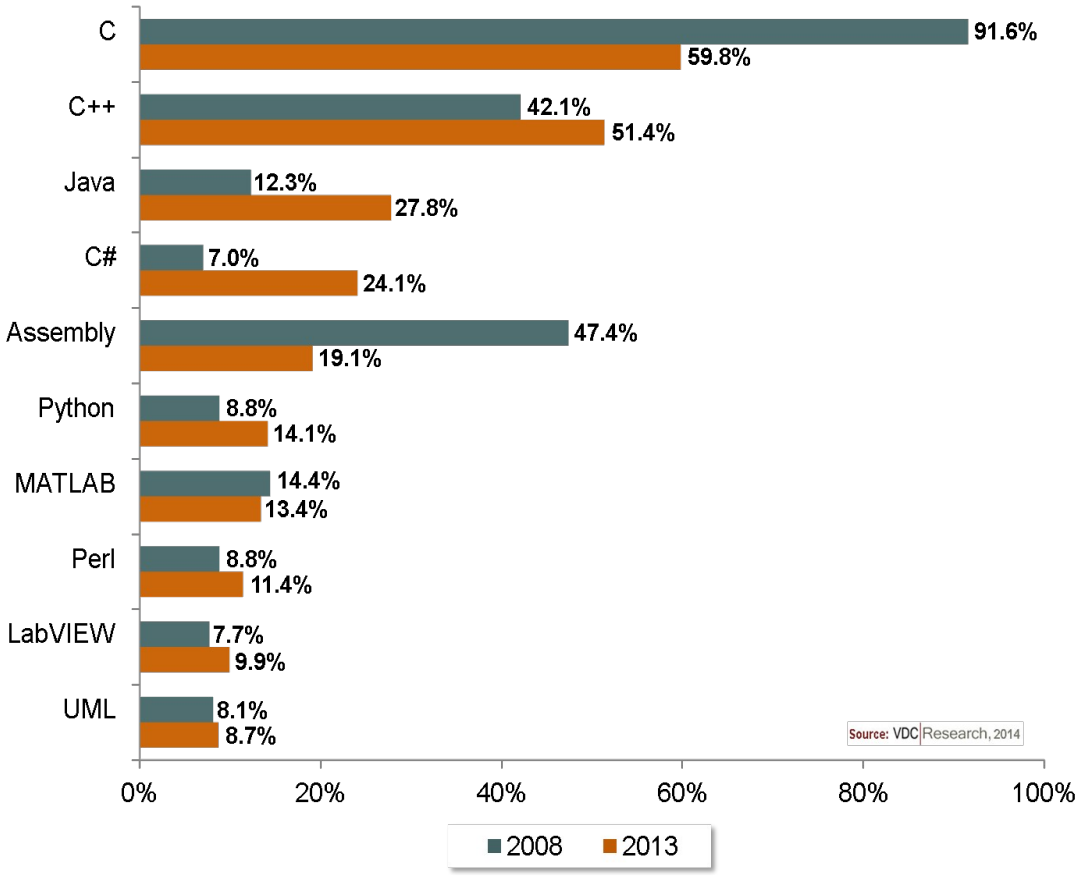
Even before increasingly ubiquitous connectivity began capturing the imagination of seemingly every product manager and engineer, a substantial transformation of the embedded industry was already underway – itself setting the stage for and enabling the rapid evolution of the IoT. Over the course of the past decade, embedded software gradually emerged as the primary vehicle through which engineering organizations could differentiate their products and control end-user experiences. The significant shift in the embedded operating system landscape epitomizes this trend, with now only 15% of engineers reporting designs without a formal operating system and almost 70% citing the use of a third-party solution (commercial or open source).

Exhibit 1: Operating system use on current projects (percent of respondents)



Unrelenting time-to-market and software content creation pressures have driven engineering organizations to fundamentally re-evaluate their software engineering processes and technologies. In this light, one of the elemental mechanisms with which software engineers ultimately shape software application development is the programming language. This fundamental medium of differentiation demonstrated a rapid shift over just the past five years as more engineers move away from embedded industry stalwarts C and Assembly in favor of object-oriented languages such as Java. As the pressures driving engineering organizations to re-evaluate incumbent technologies grow more acute, flexible and scalable solutions that can also serve as platforms for IoT-related services delivery will be placed at a premium.

Exhibit 2: Languages using to develop software on the current project *(percent of respondents)*



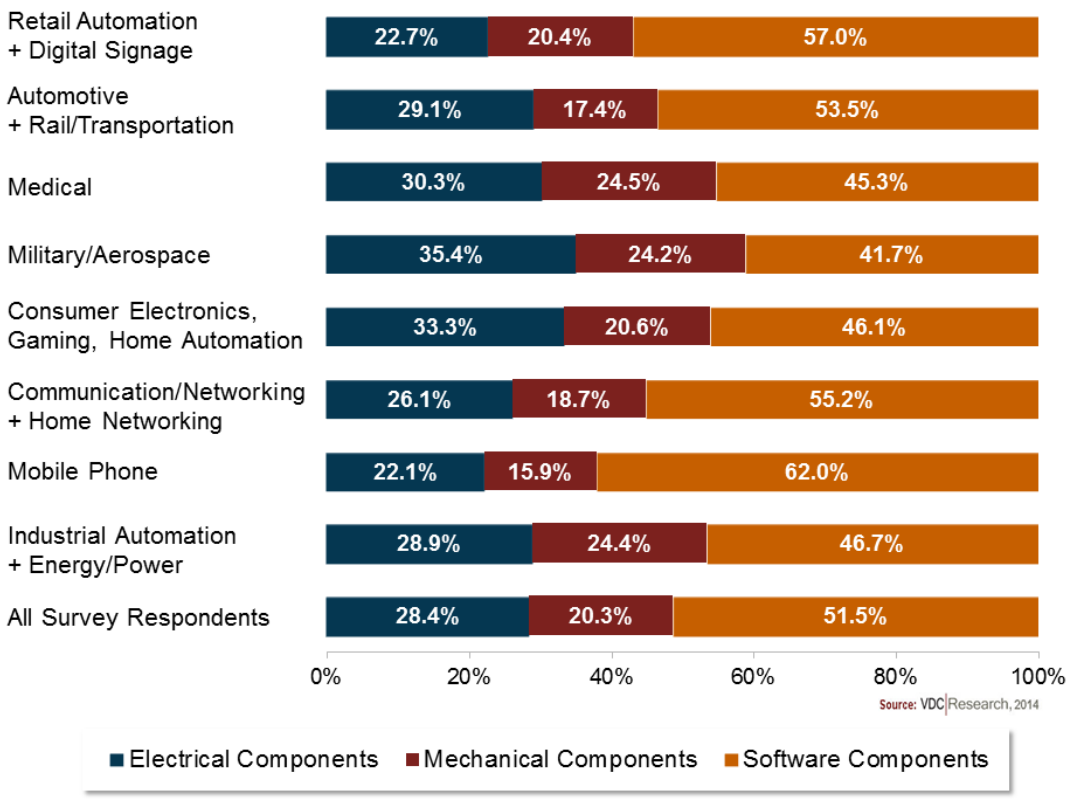
Background on VDC Research

VDC has been covering the embedded systems market since 1994. The findings from VDC's 2013 Embedded System Engineering Survey, capturing the input from 400 engineers, offer insight into leading business and technical trends impacting engineering organizations as well as the best practices implemented to address them. The respondents are directly involved in software and systems development across a range of industries including automotive, aerospace and defense, telecoms, medical, industrial automation, and consumer electronics, among others.

The Shifting Foundation of Value Creation and Differentiation

Not only has software evolved into a primary vehicle for device differentiation, but it also emerged as the primary labor cost for embedded engineering organizations. Our surveys of leading OEMs confirm that, in many instances, these organizations now dedicate more than 50% of their development effort to software – a trend that holds true across industries. This increasing burden has pushed many organizations to abandon traditional development practices that are overly rigid, inefficient, and cannot scale.

Exhibit 3: Estimated distribution of development costs on project *(average development costs)*



As engineering organizations strive to adapt, the profile of internally developed code bases is changing. Legacy code assets have been placed at a premium as more OEMs prioritize reuse. Meanwhile, increasing memory resources, multicore processors, new connectivity requirements, and unrelenting time-to-market pressures are driving developers to change programming languages for net new code development. Embedded engineering is no longer synonymous with small footprint, fixed-function devices. The requirements for high reliability and determinism that perpetuated the use of C, while still present, have now been subjugated in favor of object-oriented and higher-level languages such as C++ and Java (See Exhibit 2). However, the growing utility and application of these new software development languages will not cannibalize the need for older or incumbent languages. Many of the same pressures that are driving engineering organizations to consider alternative programming choices are also compelling them to maximize their use of existing assets. Thus, while we expect more engineering organization to recognize the value proposition of object-oriented languages with extensive libraries such as Java, the composition of tomorrow's code bases will be more diverse, not less.

This same characterization of heterogeneity is echoed in the growing use of third-party software. Not only must OEMs inject a new level of rigor into their own software development efforts, but they also must account for the increasing volume of IP

in their devices sourced from other entities. Already more than 60% of project software code is derived from third-party sources (whether commercially licensed or obtained from public sources). This ratio will only increase over time as engineering organizations contend with relentless time-to-market pressures and a need to focus their own talent on increasingly narrow bands of differentiating functionality. The widespread and growing adoption of standards-based technology only make this trend more acute.

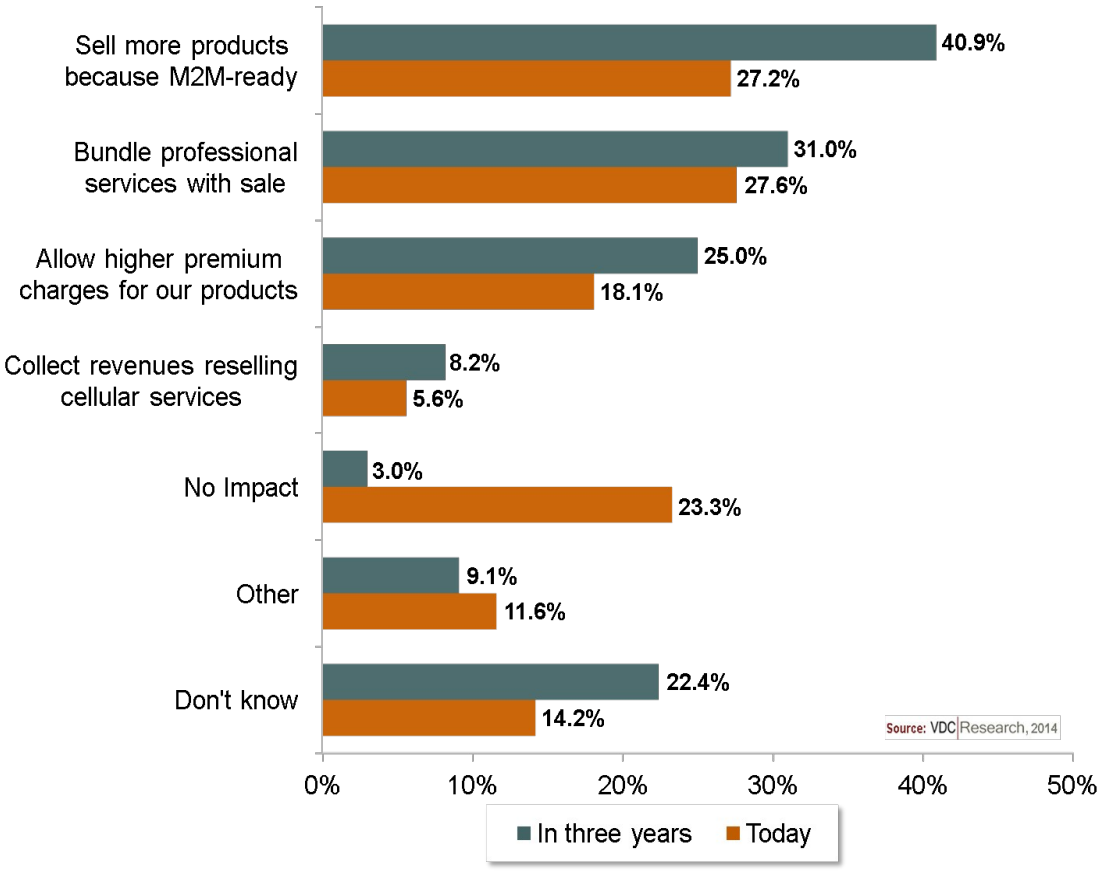
Exhibit 4: Networking capabilities/features in project *(percent of respondents)*

	All Survey Respondents		Industrial Automation + Energy/ Power		Communication/ Networking + Home Networking		Consumer Electronics, Gaming, Home Automation		Military/ Aerospace		Medical		Automotive + Rail/ Transportation		Retail Automation + Digital Signage	
	Today	In 3 years	Today	In 3 years	Today	In 3 years	Today	In 3 years	Today	In 3 years	Today	In 3 years	Today	In 3 years	Today	In 3 years
Wireless networking capabilities	53%	65%	50%	67%	59%	52%	67%	61%	44%	59%	48%	77%	41%	51%	42%	85%
Wired networking capabilities	46%	49%	62%	65%	61%	57%	39%	47%	56%	63%	29%	32%	51%	54%	27%	31%

Source: VDC Research, 2014

Connectivity and machine-to-machine communication are not new concepts within the embedded marketplace. Applications such as driver-assist telematics systems and connected factory installations have been around for more than a decade. Already a large percentage of embedded products produced support some form of wired or wireless connectivity. And while the progress against this binary categorization will continue in the coming years, the industry's real transformation will manifest itself not through the number of connected nodes but through the richness of functionality and business change being enabled through those connections. For example, nearly 20% of respondents stated that their company intends to offer usage-based product licensing within the next three years – a two-fold increase from current values (See Exhibit 5).

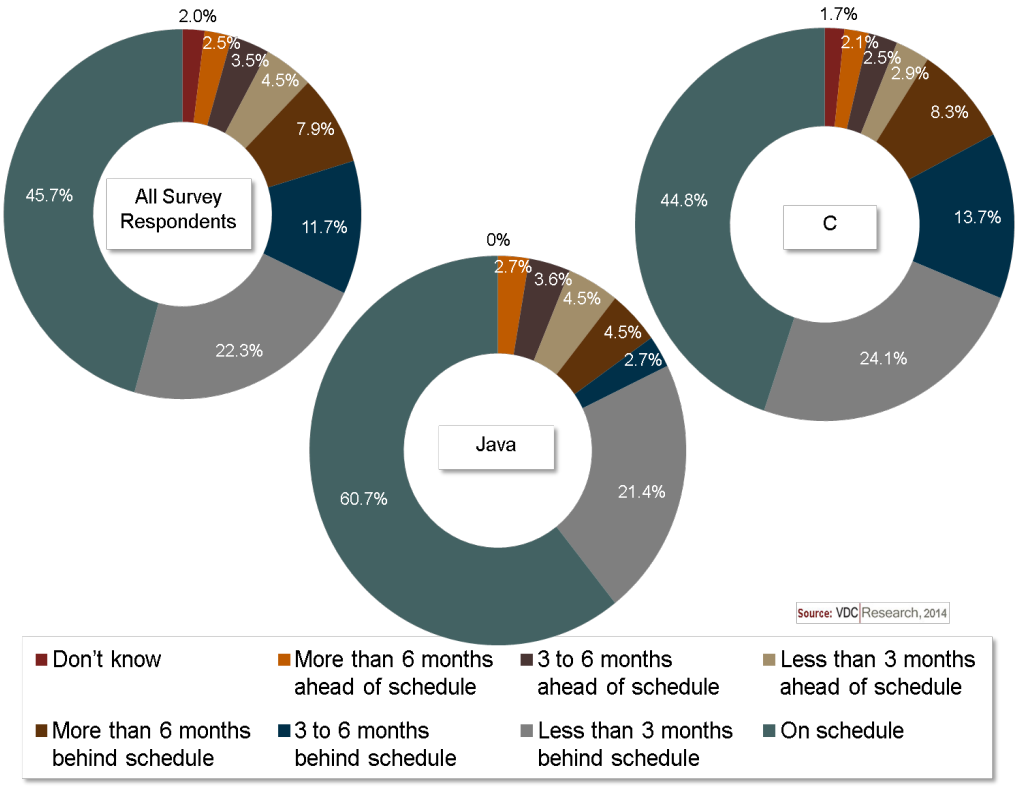
Exhibit 5: Ways the IoT/M2M is currently impacting organization's business model *(percent of respondents)*



An Industry Conditioned to Fail?

Despite the opportunities presented by the IoT, next-generation system development also introduces new layers and multipliers of complexity into already strained product development processes. To a certain extent, the complexity inherent in embedded system design has conditioned the industry to failure. In 2013, more than 40% of surveyed engineers reported projects that were behind schedule – a rate that has remained relatively consistent for the two decades that VDC has been covering the market. Furthermore, any deviation from schedule can cause a significant ripple effect of additional delays in the embedded market. Not only are embedded engineering projects much longer than most within the IT realm, but they also often involve the coordination of dependent work conducted across multiple engineering disciplines (software, mechanical, electrical, etc.). While schedule delays may not always signal abject project failure, they invariably lead to undesirable development cost overruns and lost revenue opportunities.

Exhibit 6: Current project's schedule *(percent of respondents)*



Unfortunately, the cost of delay in the embedded market has never been greater. Time-to-market windows are shrinking. Engineering project timelines that traditionally required three to five years are now being compressed to 12-to-18 months. This accelerated project execution, however, comes at a time when engineering organizations are already contending with new challenges and levels of complexity not previously encountered. The IoT is forcing OEMs to respond to rapidly changing customer requirements while also supporting and evangelizing new business models and product use cases. Clearly, the status quo no longer suffices, but new technology adoption is already offering some hope. For example, respondents using higher-level programming languages like Java, which helps promote code reuse, reported better schedule adherence than those using C – despite also having larger, more complex projects.

Exhibit 7: Way in which total lines of code is expected to change on next embedded project
(percent of respondents)

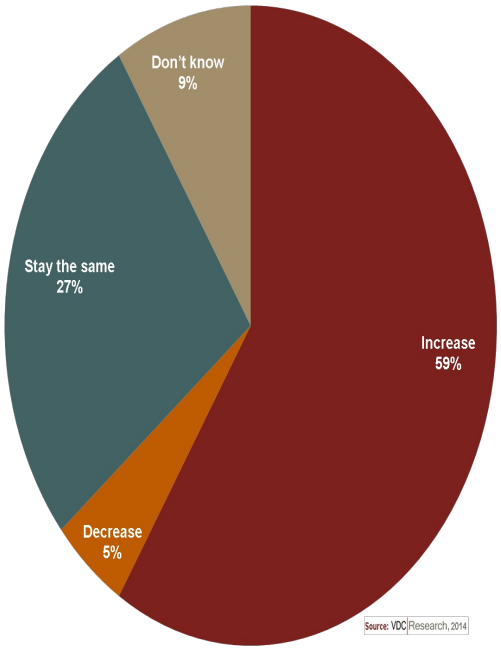
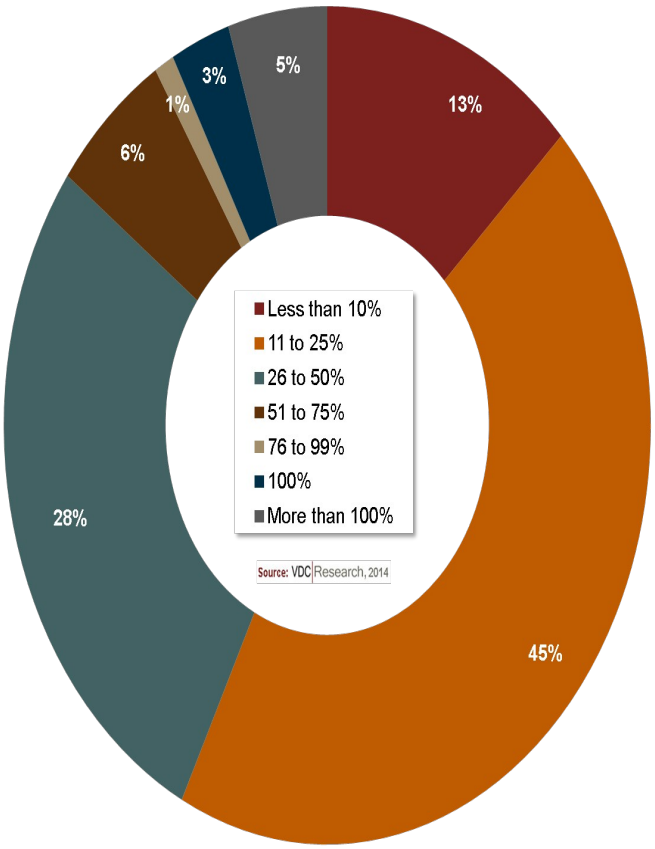


Exhibit 8: Percentage that lines of code is expected to increase on next embedded project
(percent of respondents)


13%
45%
28%
6%
1%
3%
5%
Less than 10%
11 to 25%
26 to 50%
51 to 75%
76 to 99%
100%
More than 100%
Source: VDC Research, 2014

In fact, respondents to our engineering survey expect code base sizes from each of those sources to grow at rates far outpacing potential increases in productivity from internal resources. Unfortunately, the worldwide population of embedded engineers cannot grow fast enough to support these requirements either. The pipeline from higher education is too small. Our research predicts that this community will grow at only 6.9% through 2016. The operational challenges associated with the skilled labor shortage will then be further magnified by the subsequent premium cost placed on those resources that are available. Tools and process efficiencies can help, but an even greater reliance on third-party software should be expected. As such, embedded engineering organizations must also evaluate new resource pools to address the growing software content creation requirements. This evaluation will likewise place a growing premium on skill sets such as Java that are transferable between the enterprise and embedded domains.

In parallel to the emergence of the aforementioned resource gap, the evolution of embedded devices is making IT developer skill sets increasingly relevant. Traditionally, the heterogeneous landscape of hardware and runtime components within the embedded marketplace created a siloed, inelastic, and somewhat limited job market. Now, however, the increasing amount of on-chip resources and evolving end-device functionality requirements have fueled the expansion of experimentation with enterprise/IT class technology within enterprise designs.

The impact of next-generation connected systems is not limited to new device software stacks. Today's devices are now producing an exploding amount of operational and environmental sensor data. This new content requires technology to support both localized native decisions as well as an intelligent infrastructure to support integration with IT systems and analytics databases. This new dynamic is one for which in-house run-times were just not designed, causing even more churn in technology evaluation. As the associated data management issues become a greater issue, engineering organizations and enterprises will be forced to look for new solutions capable of storing and transmitting data, which was not a consideration for most OEMs in the past.

The IoT Unifies Fragmented Embedded Ecosystem, Driving Need for Scalability

For years, fragmentation was the unifying theme within the embedded market. The industry was an amalgamation of distinct device types that each possessed its own heterogeneous ecosystem of supporting components. However, the Internet of Things has now catalyzed (or at least accelerated) a convergent evolution of these various system types. This new ecosystem has codified around a new theme and horizontal objective – to deliver rich, contextually aware functionality and post-deployment, value-add services. For example, OEMs historically chose from dozens of different operating systems, many of which were designed to meet specific sets of resource constraints or performance requirements (e.g. real-time latency, footprint, lack of MMU, or industry-specific middleware feature sets). Over time, however, general-purpose and open source OSs such as Linux garnered increased adoption in the embedded marketplace, gradually displacing entrenched perceptions of the run-times' utility. Fewer engineering organizations can afford to invest their internal labor resources in differentiation below the application level.

In addition to the growth of software content, the sophistication and complexity of today's hardware platforms are having a tremendous impact on system engineering considerations. In many ways, the commoditization of compute power has been the engine underlying the new applications in the IoT era. There are simply far more relevant embedded processor families and architectures than within the traditional IT and server landscape. Not only are today's semiconductor and hardware platforms becoming more complex, but the diversity, number, and structure of processing elements within the average hardware design is evolving as well. Although there has been consolidation around ARM and x86 processors in recent years, the sheer number of chipsets and the increasing pace of product evolution present challenges. For example, more engineering organizations are adopting heterogeneous system architectures, composed of a diverse set of processor types and peripherals, to cost-effectively develop more application-optimized systems.

Exhibit 9: Processing units used on current project *(percent of respondents)*

Processing unit(s) used on current project *(Percent of Respondents)*									
	All Survey Respondents	Industrial Auto + Energy/ Power	Mobile Phone	Communication/ Networking + Home Networking	Consumer Electronics, Gaming, Home Automation	Military/ Aerospace	Medical	Automotive + Rail/ Transportation	Retail Automation + Digital Signage
MCU *(MicroController Unit)*	52%	67%	39%	34%	36%	59%	68%	76%	38%
MPU (MicroProcessing Unit)	44%	39%	54%	59%	33%	53%	45%	35%	38%
FPGA *(Field Programmable Gate Array)* or **PLD** *(Programmable Logic Device)*	30%	31%	23%	20%	14%	69%	26%	24%	23%
GPU *(Graphics Processing Unit)*	29%	24%	53%	32%	39%	25%	16%	19%	27%
SoC *(System on Chip)*	25%	20%	28%	30%	39%	19%	19%	14%	15%
DSP *(Digital Signal Processor)*	22%	32%	33%	11%	36%	25%	6%	22%	4%
System relies on external/ distributed processing	12%	5%	16%	14%	8%	13%	3%	5%	15%
Custom ASIC *(Application Specific Integrated Circuit)*	10%	2%	7%	18%	19%	3%	6%	14%	15%
MPPA *(Massively Parallel Processor Array)*	6%	8%	9%	7%	14%	3%	3%	0%	8%
Other	1%	0%	0%	0%	0%	3%	3%	0%	0%

Processor family of Primary processor *(Percent of Respondents)*									
	All Survey Respondents	Industrial Auto + Energy/ Power	Mobile Phone	Communication/ Networking + Home Networking	Consumer Electronics, Gaming, Home Automation	Military/ Aerospace	Medical	Automotive + Rail/ Transportation	Retail Automation + Digital Signage
ARM Cortex *M/SecurCore/ ARM9T/ARM7*	21%	24%	21%	18%	19%	22%	35%	16%	12%
x86	19%	26%	19%	16%	17%	25%	13%	14%	19%
ARM Cortex *A/ARM11*	9%	5%	16%	9%	11%	0%	3%	8%	8%
ARM Cortex *R/ARM10/ARM9E*	6%	4%	5%	5%	8%	13%	6%	3%	15%
MIPS32/16e/ microMIPS	6%	2%	2%	5%	11%	0%	13%	11%	15%
MIPS64	6%	1%	18%	7%	6%	9%	3%	3%	0%
PowerPC	5%	2%	2%	11%	0%	9%	0%	14%	8%
MSP430	4%	8%	4%	2%	0%	6%	10%	3%	4%
AVR	3%	5%	2%	7%	0%	0%	3%	3%	8%
Other	20%	23%	12%	20%	28%	16%	13%	27%	12%
TOTAL	100%	100%	100%	100%	100%	100%	100%	100%	100%

Source: VDC Research, 2014

These new processing platforms, which are often multicore and/or multiprocessor in design, can impose a new set of development challenges for embedded software engineers with experience mostly gained implementing serial applications. Now developers must architect applications to allow for thread or task parallelization (while maintaining any needed determinism) and account for new sets of potential run-time errors, such race conditions and deadlocks. So not only are engineering organizations facing pressure to bring new IoT-ready products to market, but they also must do so within programming paradigms that still remain rather foreign. Going forward, these growing software development challenges will only reinforce the adoption trends of languages such as Java that are better tailored for parallel programming models than the likes of C.

The proliferation of small form-factor devices, from wearable consumer electronics to unmanned aerial vehicles, will unavoidably perpetuate the heterogeneous processing landscape we see today as OEMs look for power- and footprint-optimized architectures. The elasticity of today's software and service functionality requirements, however, necessitate greater corporate agility than traditionally afforded by the industry's associated roster of proprietary software platforms. As such, the ability to develop applications that are portable and scalable across architectures, as facilitated by advances in technology such as Java 8 (which now offers more compact profiles and the ability to create custom runtimes), will be placed at a premium as engineering organizations look for migration paths to standards-based technologies.

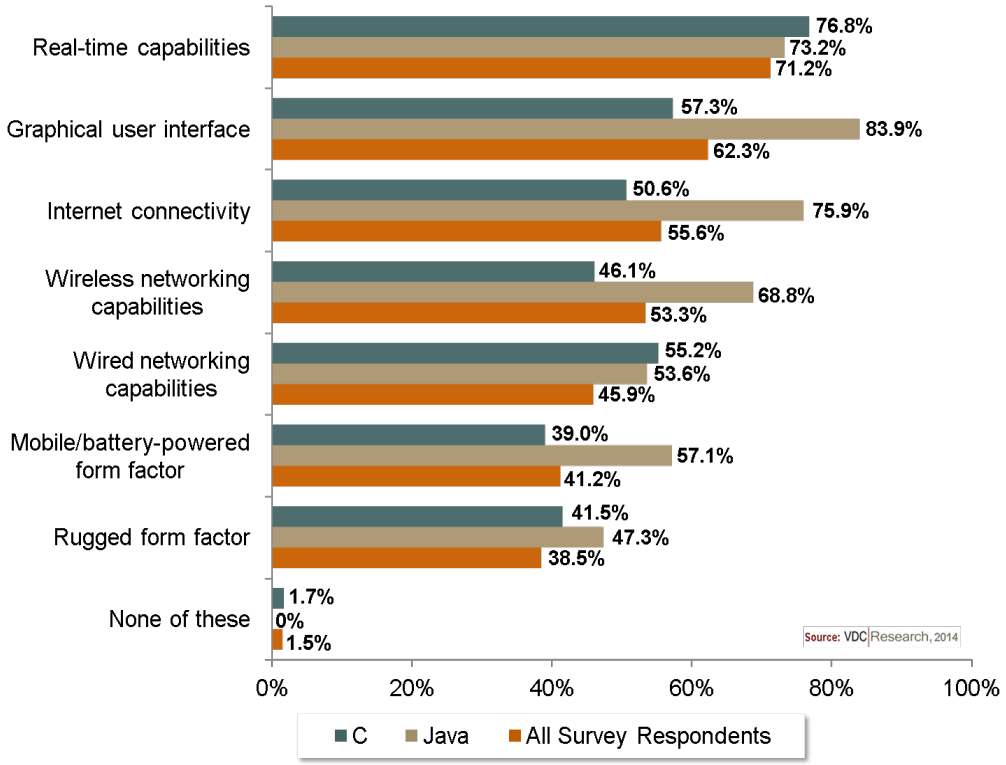
OEMs See an Object-Oriented Future

Traditionally, embedded software development was predominantly conducted in C. As discussed above, however, embedded engineering is no longer synonymous with small footprint, fixed-function devices. Just as developers once migrated from Assembly to C, OEMs are already embracing alternatives. The requirements for high reliability and determinism that perpetuated the use of C, while still present, have now been subjugated in favor of object-oriented and higher-level languages such as C++ and Java that offer useful features for more sophisticated systems design, such as advanced memory management models and libraries catering to GUIs. The magnitude of this change over just the previous five years is tremendous given traditional pace of evolution in embedded. The rate of this change speaks to overall acceleration in rate of innovation and new feature set requirements in new product development.

The traditional barriers and reservations about using Java in embedded devices are wearing down, too. For one, memory requirements are less of an issue given the resources found within today's embedded hardware landscape. The introduction of Lambdas in Java 8 also provides a way for engineers to develop software more efficiently with less boilerplate code. Additionally, the Lambdas enable developers to more easily distribute the processing of collections over multiple threads on multi-core processors, resulting in better application performance and simpler code. Furthermore, enhancements to the Java embedded portfolio now present a more compelling solution that, with footprint as small as 128kb for ME Embedded 8, scales down to small ARM-based devices and up to x86-based datacenters. Now, the Java Platform, Standard Edition (Java SE) and Micro Edition (Java ME) libraries and roadmaps are much more closely aligned and the SE platform offers a level of modularity – and footprint customization – not previously possible.

Additionally, the concerns over Java's suitability for mission- or safety-critical applications with deterministic latency requirements are dissipating. A majority of surveyed embedded Java users now report using Java in projects with real-time requirements. In some cases, engineers deploy mixed-language systems or create custom hardware to offload or accelerate response time for certain functions. However, various other initiatives such as RTSJ (Real-Time Specification for Java) and third-party vendor solutions (e.g. aicas, Atego, or IBM Websphere) have helped embedded engineering organizations identify ways to mitigate some of the latency concerns revolving around garbage collection for deterministic threads. Also of note, the profile or available resources on today's devices are vastly different than those of prior generations. Now, organic bill-of-material changes, driven by decreasing cost of system memory and increasing mips/watt processor performance, are also removing some of that need for concern. As the level of experience working with Java increases within the embedded community and the platform's hardware support continues to diversify, the platform's growth trajectory will only be reinforced.

Exhibit 10: Capabilities/features included in current project *(percent of respondents)*



Conclusion

The Internet of Things is rendering many incumbent embedded engineering technologies and design processes insufficient and antiquated. Already, the growing role of software to electronic system functionality and differentiation had upturned an ecosystem traditionally averse to change. Although many of the requirements for embedded devices remain unique and often industry-specific, others such as connectivity, enterprise integration, security, and data analytics are becoming more important.

The increasing number of interconnected embedded devices will place an even greater premium on the ability to effectively manage continual content distribution – whether for updates, upgrades, or managed services. The proportion of software functionality and end-user value delivered post-deployment will magnify any existing software development inefficiencies. OEMs will face mounting pressure to design and deliver software as a seamless part of the overall system architecture and product lifecycle, creating an infrastructure for continuous engineering and differentiation evolution. Engineering organizations now need new solutions that address these evolving requirements and speed development and time to revenue. In many cases, proven, enterprise-grade technologies such as Java are re-emerging as increasingly relevant and compelling solutions within this IoT ecosystem.

VDC Research
Insights for the Connected World

Market Intelligence for Technology Executives. VDC Research Group (VDC) provides market research and advisory services to the world's top technology executives. Our clients rely on us to provide actionable insights to support their most important strategic decisions. The firm is organized around four practices, each with its own focused area of coverage including: automatic identification and data collection, embedded hardware, embedded software and enterprise mobility.

For more information about this research, please contact:

VDC Research Group, Inc. | 679 Worcester Road | Suite 2 | Natick, MA 01760

508.653.9000 x123, info@vdcresearch.com

VDC research

Chris Rommel – *Vice President, M2M Embedded Technology*