

THE FOLLOWING IS AN EXECUTIVE WHITE PAPER ON:

EMBEDDED OPPORTUNITY: DYNAMICS OF MEDICAL DEVICE EVOLUTION DRIVING NEED FOR STATIC ANALYSIS

Licensed to Distribute by:
Klocwork, Inc.

Prepared by:

Steve Balacco
Director, Embedded Software and Tools Practice

Chris Rommel
Analyst, Embedded Software and Tools Practice

July 2009
All Rights Reserved

EMBEDDED OPPORTUNITY: DYNAMICS OF MEDICAL DEVICE EVOLUTION DRIVING NEED FOR STATIC ANALYSIS

Evolving end-user expectations for differentiated software functionality, sustained time-to-market, and cost pressures continue to fuel new software development within embedded devices. Time-to-market and cost considerations will continue to drive embedded systems manufacturers to utilize international engineering resources and legacy code assets whenever efficient and economical. The rising complexity of the associated design requirements and software code bases will force more manufacturers to consider new testing methodologies and tools in order to adequately assess software quality and deliver reliable products to market as quickly as possible.

The medical device industry, in particular, presents an intriguing paradigm within the embedded engineering landscape where many of the overarching trends affecting the broader embedded domain are magnified and accelerated. The unparalleled predicated rate of software code growth within these devices is juxtaposed to development teams that are smaller and have fewer software and test engineers as a percent of team size than those within other embedded device industries.

EMBEDDED SOFTWARE CODE GROWING AND DIVERSIFYING

The evolution of semiconductor technologies has removed many of the resource constraints associated with traditional embedded devices, allowing for more robust operating systems and more sophisticated run-time applications. In conjunction with these technological advancements, the once finite boundaries between embedded and desktop computing have blurred as PC and embedded manufacturers strive to meet the evolving customer demands for enhanced software functionality, mobility, and connectivity.

In order to enable the additional features and functionality dictated by the marketplace, the number of lines of code found in embedded devices has continued to grow to where the mean embedded project now has over 130,000 in-house developed lines of software code in addition to over 200,000 commercial/third-party lines of code with many larger projects having millions of total lines of code. Although there is wide range in the amount of software code typically included across the numerous embedded application classes – all devices will be expected to do more, faster, with less power, and often with a more sophisticated and intuitive user interface.

Number of Lines of Software Code in Final Design
(Mean of Respondents)

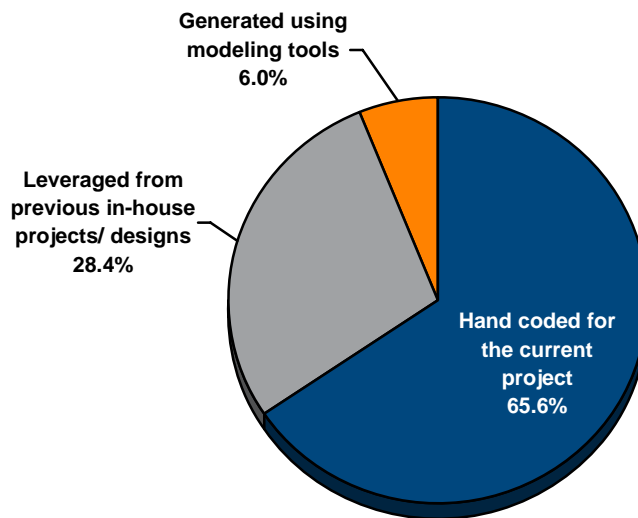
Number of Lines of Software Code in Final Design (Mean of Respondents)	
Commercial/Third-Party Lines of Software Code	<u>Aggregate</u> 204,645
In-house Developed Lines of Software Code	130,443

The once finite boundaries between embedded and desktop computing have blurred as PC and embedded manufacturers strive to meet the evolving customer demands for enhanced software functionality, mobility, and connectivity.

The number of lines of code, however, is only one metric to measure the growing complexity of the software content in embedded devices. The number and quality of other sources of code being used within a project also contributes to system complexity and amplifies the importance of rigorous automated testing.

On average, nearly 30% of in-house developed software code for current embedded projects is reused from previous projects or designs. This software code represents enormous investments in previous engineering efforts and in an economic climate when reduced resources are common these assets can be critical to sustaining budgets and time-to-market deadlines.

Source of In-house Developed Software Code for Current Project
(Mean of Respondents)



However, the original architects and developers of this legacy software will have often left the company, which places additional risks on any updates or integrations performed with the reused code. So some of the dangers typically associated with integrating third-party commercial or open source software are actually indirectly applicable to code that a company believes to be tried and field-proven. To help them manage the sustained quality and functionality issues related to the ever-growing size of code bases and the reuse of legacy code, engineering teams will need to continue to search for new tools and methodologies.

DISTRIBUTED ENGINEERING NECESSITATES FOCUSED Q/A

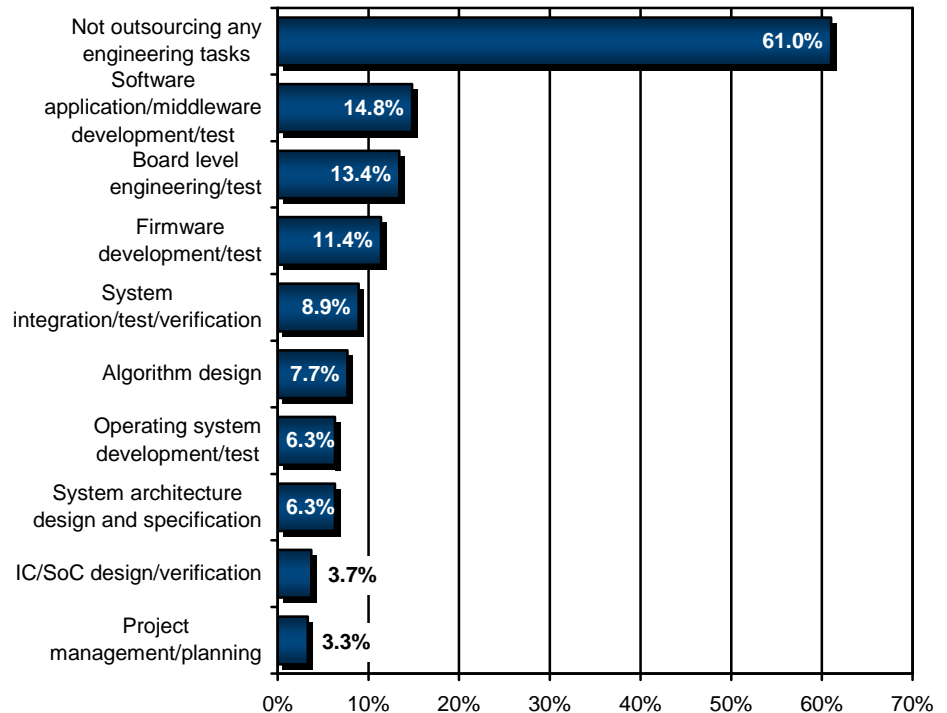
The globalization of the economy has allowed businesses in all domains to expand their worldwide footprints and establish multinational workforces that can take advantage of the best talent as well as the lower-cost labor markets from around the world. The recent recession has amplified this unrelenting need for strategies to drive efficiencies and reduce operational expenses.

Embedded systems manufacturers began embracing the use of international labor markets and third-party engineering resources long before the recent economic events began driving organizations to look for new ways to reduce costs. VDC's 2008 Embedded System Engineer Survey indicates that almost 40% of projects are outsourcing some engineering tasks to external companies. Moreover, 12.5% of surveyed engineers reported that the majority of their current project team was located in other countries.

Despite any incentives to promote the rehabilitation or expansion of domestic economies, outsourcing will remain one of the critical strategies for companies to ensure short-term agility and long-term viability.

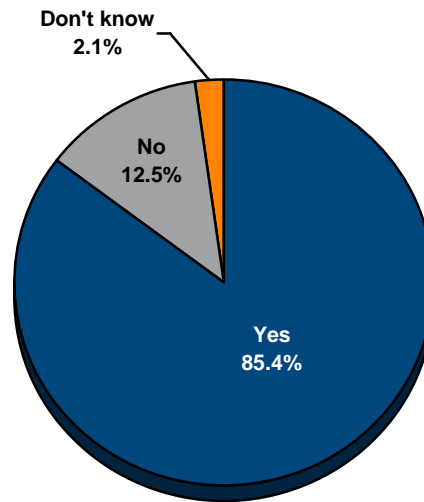
The high costs of labor and benefits within the United States and Western Europe will continue to drive OEMs to utilize the rapidly maturing expertise available through third-party engineering services companies. Despite any incentives to promote the rehabilitation or expansion of domestic economies, outsourcing will remain one of the critical strategies for companies to ensure short-term agility and long-term viability.

Engineering Tasks Outsourced to External Companies for the Current Project
(Percent of Respondents)



Note: Percentages sum to over 100% due to multiple responses.

Majority of Current Project's Engineering Team Located within Same Country as Respondent
(Percent of Respondents)



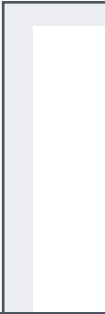
Difficulties in accurately articulating the design requirements can introduce unneeded errors into components of the software code while challenges in synchronizing builds across time zones and continents can also augment these issues.

As a result of this high level of diversity within their project teams, embedded device developers are also forced to negotiate the added complexity wrought by varying business hours, native languages, and expertise as they are already attempting to meet increasingly complicated design requirements. Difficulties in accurately articulating the design requirements can introduce unneeded errors into components of the software code while challenges in synchronizing builds across time zones and continents can also augment these issues. VDC expects that more engineering teams will begin evaluating and adopting new development tools and methodologies in order to address the challenges associated with geographically dispersed teams.

DON'T LET SOFTWARE ERRORS FLATLINE FUTURE PROFITABILITY

The Emerging Productivity Challenge for the Medical Device Sector

The dynamics within the medical device industry present an especially clear illustration of the key trends and issues affecting embedded system manufacturers and engineers. The increasing demands for more advanced graphics, user interfaces, and mobility is fueling rapid development of more sophisticated next generation medical devices. Moreover, the incorporation of advanced networking capabilities that can automate interfaces with patient records and other critical IT systems will place even greater stress on the development of this software content.



Furthermore, the United States Census Bureau estimates that 13% of the 2010 population will be 65 years old or greater. This demographic is expected to reach nearly 20% of the US population by 2030.¹ This dramatic transformation will fuel increasing need for sophisticated devices that can facilitate the efficient management, care, and comfort of an aging population. The development of these next generation medical devices, however, is under increasing scrutiny as a growing portion of their functionality – and differentiation – is being driven by embedded software.

The exploding level of software content, connectivity, and sheer volume of these devices will necessitate that manufacturers vigorously reevaluate their software development and testing practices.

The FDA reports that there have been 435 medical device recalls since January of 2003 based on software errors. Although the safety-critical nature of these defects varies greatly, they serve as a salient reminder of the growing amount, importance, and potential consequences of embedded software to the healthcare industry. The exploding level of software content, connectivity, and sheer volume of these devices will necessitate that manufacturers vigorously reevaluate their software development and testing practices.²

PLANNING TO FAIL?

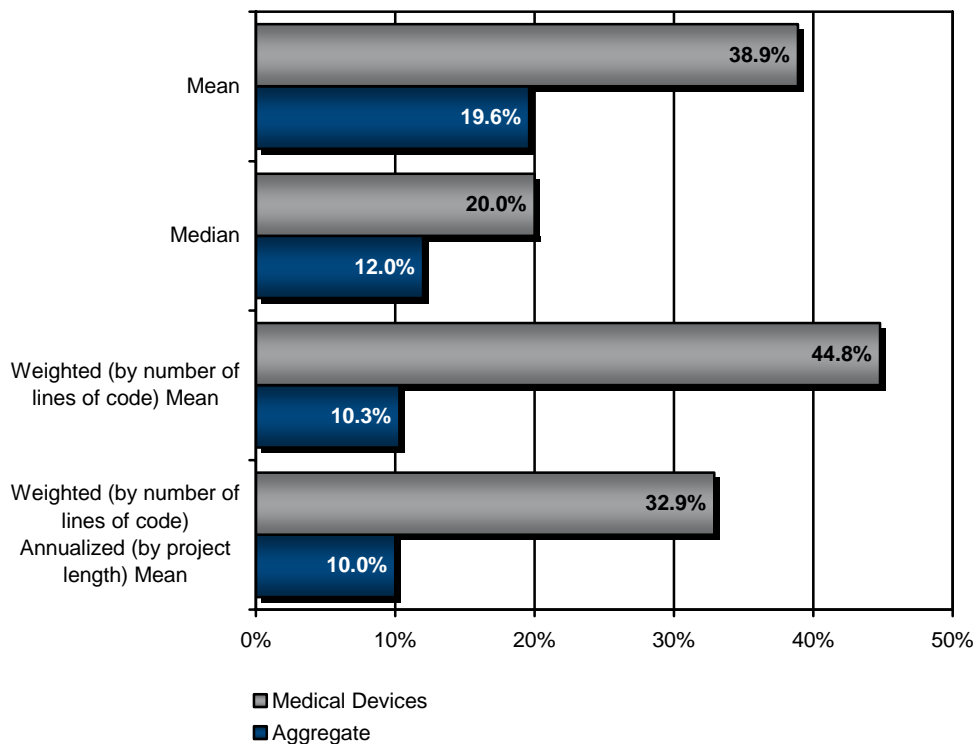
Medical Device Code Growth May Expose Engineering Resource Gap

Comparatively larger code size is not always an indicator of more complex system functionality, as smaller code bases can result from superior system architecture, coding techniques, and more efficient use of system resources. However, the expected growth rate in lines of code for medical device projects dwarfs those of most other embedded device classes, indicating a substantial increase in the amount of software content to be required within the next generation of devices as a growing amount of these devices' functionality is being driven by software.

¹ U.S. Census Bureau, 2004, "U.S. Interim Projections by Age, Sex, Race, and Hispanic Origin," <http://www.census.gov/ipc/www/usinterimproj> (accessed June 5, 2009)

² U.S. Food and Drug Administration, "CDRH Device Recalls Database," <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRES/res.cfm> (accessed June 5, 2009)

Expected Increase/Decrease in the Percent of Lines of Software Code on the Next Project
(Average of Respondents)^{3 4}



Based on developers' expectations for increases in lines of software code and software engineering effort, there is a substantial disconnect between current software engineering resources and the growth in software content on the horizon.

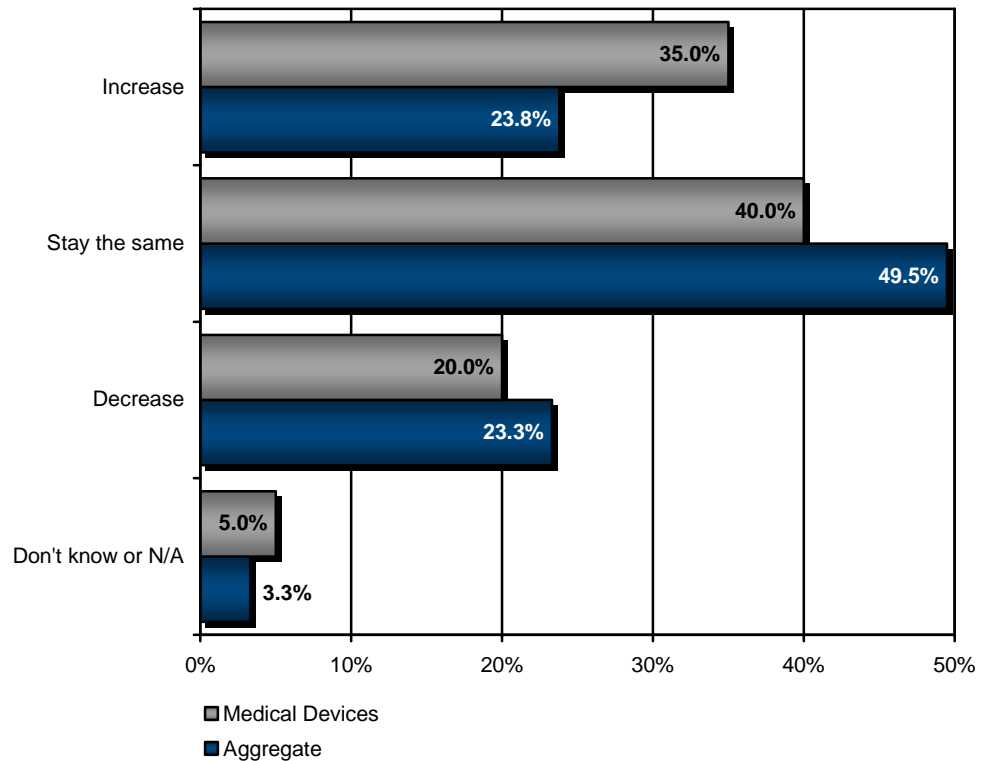
Within the medical device industry only 29.6% of the average engineering team is comprised of software engineers, compared to 41.1% across all device classes. In addition, the percentage of medical device engineering teams dedicated to test, verification, and validation is roughly half that of overall embedded industry. Based on developers' expectations for increases in lines of software code and software engineering effort – which both drastically outpace that of the aggregate sample – there is a substantial disconnect between current software engineering resources and the growth in software content on the horizon.

Engineering Team Size Working on Current Project (Average of Respondents)				
	Aggregate		Medical Devices	
	Value	% of Total	Value	% of Total
Project managers	2.6	9.3%	1.9	9.2%
Software engineers	11.6	41.1%	6.1	29.6%
Hardware engineers	6	21.1%	6.8	33.0%
System architects	1.7	6.1%	2.1	10.2%
Algorithm developers/functional experts	1.8	6.4%	2	9.7%
Test/verification/validation engineers	4.5	16.0%	1.7	8.3%
Total engineers	28.3		20.6	

³ Because respondents cite a wide range of code bases sizes in addition to the expected growth rate, the *Weighted Mean* provides a metric for the overall expected code growth by weighting each response by the respondent's current code base size as compared to that of the aggregate.

⁴ Similarly, current project length varies greatly among the respondents. The *Weighted Annualized Mean* accounts for each respondent's current code base size as well as the length of their current project in order to ascertain an overall annual rate of growth for the industry's software code base.

Expected Changes in Software Application/middleware Engineering Effort
(Percent of Respondents)



Medical device manufacturers who do not invest in new resources and tools for both embedded software development and quality assurance risk losing the ability to quickly and efficiently develop and deliver quality products to market.

This engineering resource gap will likely drive many OEMs within the medical device industry to bring in additional, potentially inexperienced engineers, force current engineers to increase their own outputs, or outsource some of the engineering work to external companies – all of which can have significant adverse effects on code quality.

Given the expected rate of code growth and the safety-critical nature of many healthcare applications, medical device manufacturers who do not invest in new resources and tools for both embedded software development and quality assurance risk losing the ability to quickly and efficiently develop and deliver quality products to market.

GROWTH IN SOFTWARE CODE BASES DRIVING THE NEED FOR AUTOMATED TEST TECHNOLOGIES

VDC estimates code bases to be growing at a weighted annualized rate of nearly 10% per year based on our 2008 analysis. Although it is no surprise that IDEs, compilers and debuggers are used in high frequency across all sizes of embedded projects, our data demonstrates a significant correlation between the number of in-house developed lines of code and the use of certain lifecycle management tools as embedded engineering teams look for new ways to develop and manage these growing code bases.

As further indication of the added challenges associated with growing code bases and engineering team sizes, engineers whose current projects contained 50,000 or more lines of in-house developed code reported using requirements management tools at a rate nearly 10% greater than those engineers reporting less than 50,000 lines of code (23.8% versus 14.2%). Our analysis also demonstrated a significant correlation between the number of lines of in-house developed lines of code and the use of source/change and configuration management tools, with over 80% of developers citing 100,000 lines of code or greater compared to only 54.8% of developers with less than 50,000 lines of in-house developed lines of code in their current project.

Engineering Team Size Working on Current Project (Average of Respondents)			
	<u>100,000 or More</u>	<u>50,000-99,999</u>	<u>Less than 50,000</u>
IDEs, editors, compilers, and/or debuggers	93.3%	90.0%	90.3%
JTAG debuggers / In-circuit emulators	77.3%	73.3%	61.9%
Software/system modeling tools (UML/SysML, SDL, Simulink, LabVIEW Development Platform, etc.)	28.0%	36.7%	25.2%
Algorithm development tools (MATLAB, etc.)	20.0%	6.7%	21.3%
Software code/document generation tools	48.0%	43.3%	32.9%
Static analysis tools (GrammarTech, Klocwork, etc.)	32.0%	20.0%	16.8%
Dynamic software testing tools	13.3%	6.7%	13.5%
Instruction set simulators	8.0%	16.7%	12.9%
Virtual hardware prototyping/simulation tools (VaST, Virtutech, etc.)	8.0%	0.0%	9.0%
Requirements management tools	22.7%	26.7%	14.2%
Source/change/configuration management tools	82.7%	80.0%	54.8%
Project management tools	40.0%	50.0%	34.2%
None of the above	0.0%	0.0%	0.6%

Note: Percentages sum to over 100% due to multiple responses.

We also believe that automated test tools are often of greater importance to embedded engineering teams managing larger code bases. Although dynamic test tools demonstrate a fairly consistent rate across code size segmentations, static source code analysis tools are used by developers with 100,000 lines of in-house developed code or greater at a rate almost twice that of developers citing less than 50,000 lines of code.

Static source code analysis tools are used by developers with 100,000 lines of in-house developed code or greater at a rate almost twice that of developers citing less than 50,000 lines of code.

These growing code bases cause the potential number of executable paths within the code to increase dramatically. This, in turn, limits the more traditional dynamic test tools' ability to achieve acceptable code coverage – both in regards to the sheer number of tests needed to be executed as well as the amount of time needed to run them.

By simultaneously increasing the lines of code where bugs could appear and decreasing the practicality and effectiveness of traditional testing techniques, development teams are increasingly looking toward more formal methods of proving their software's functionality and quality.

The rising complexity of embedded software design requirements will force more manufacturers to consider new testing methodologies and tools, such as those for advanced source code analysis, in order to adequately assess software quality and deliver reliable products to market as quickly as possible.

Static source code analysis offers a way for individual engineers to test code without needing to execute it, and does so earlier in the development process than possible with unit test tools. As such, while traditional dynamic testing will remain a vital instrument within sound testing practices, the relative importance of automated static analysis to embedded device and system development should continue to increase going forward.

The current level of adoption of these tools within the embedded software development community remains relatively low, especially since much of the current use is attributable to Lint-like syntax parsers and pattern checkers and not the more sophisticated analysis tools. The rising complexity of embedded software design requirements will force more manufacturers to consider new testing methodologies and tools, such as those for advanced source code analysis, in order to adequately assess software quality and deliver reliable products to market as quickly as possible.

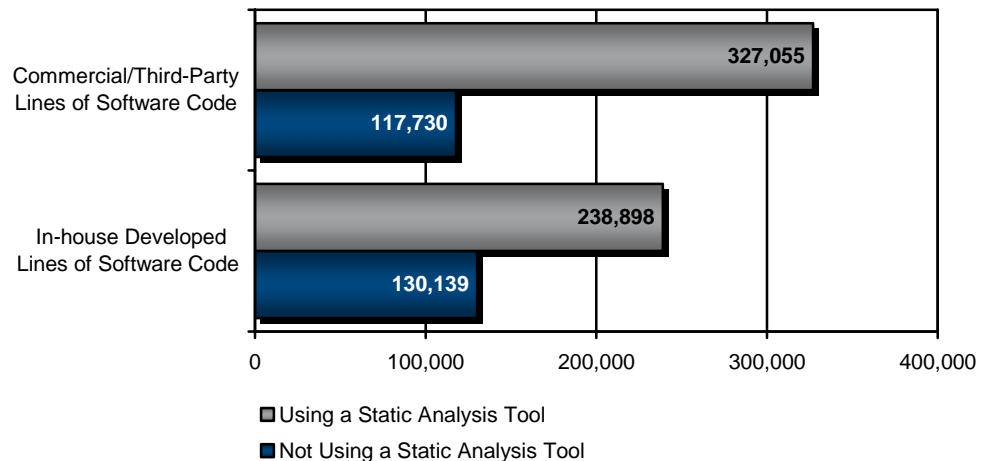
STATIC SOURCE CODE ANALYSIS ADDRESSING EMBEDDED ENGINEERING CHALLENGES

Static Analysis Users Tackle Larger Code Bases

As code bases continue to deepen and broaden, VDC expects sophisticated code analysis tools to emerge with greater importance to ensure the proper functionality of embedded software components.

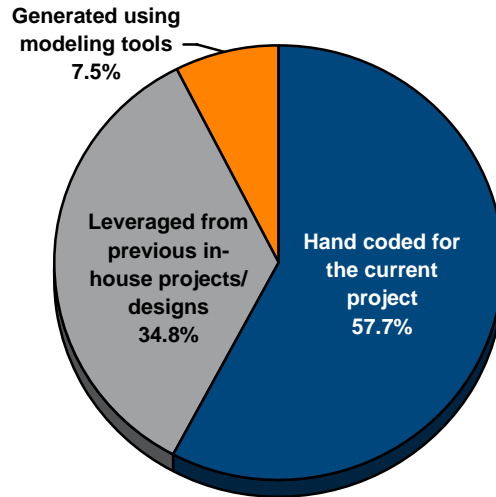
Our survey results suggest that software engineers across the embedded landscape have already begun combating the added complexity wrought from large code bases through the use of static analysis tools. This is evidenced by an average project code base size from developers already using a static analysis tool that is much larger than that of those who are not using one.

Number of Lines of Software Code in Final Design
(Mean of Respondents)

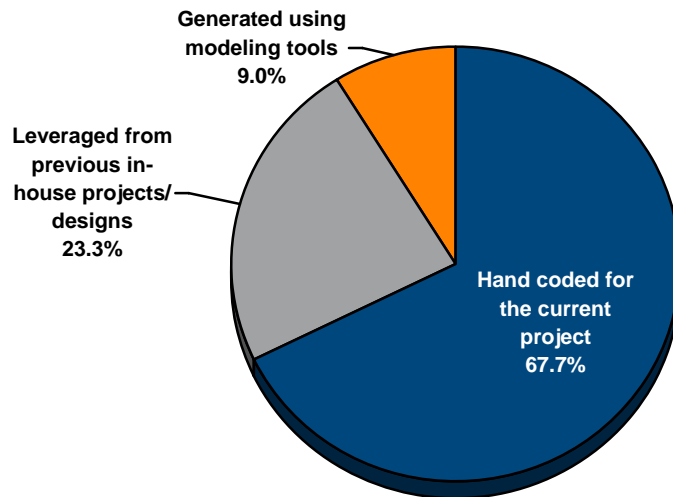


Source of In-house Developed Software Code for Current Project
(Mean of Respondents)

Using a Static Analysis Tool



Not Using a Static Analysis Tool



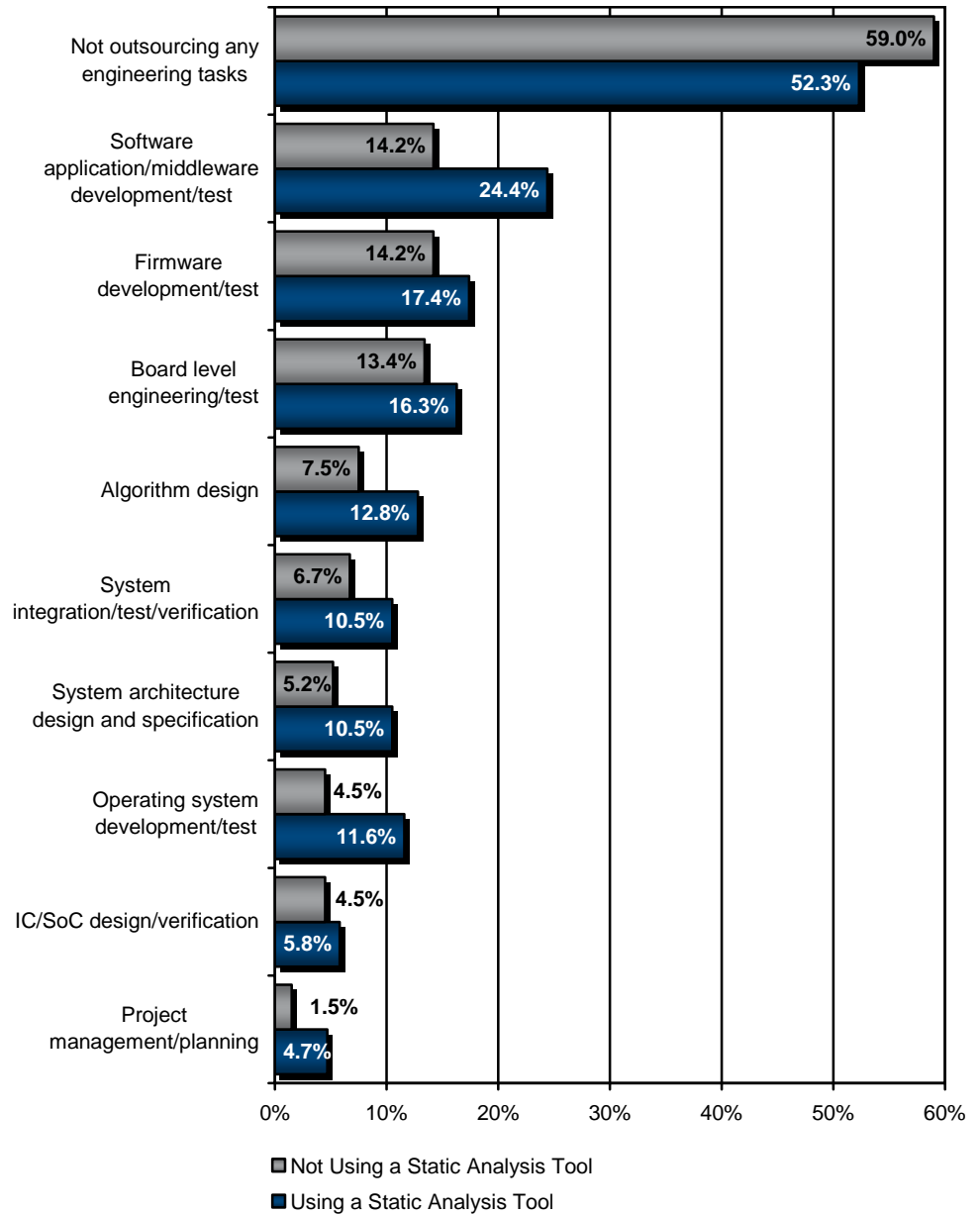
Moreover, users of static analysis tools also indicated a higher percentage of in-house developed software code leveraged from previous projects. So in addition to negotiating a larger number of commercial/third party lines of code, these engineers are also integrating larger volumes of legacy code that is potentially authored by a variety of internal sources. Clearly, embedded systems manufacturers are looking to static analysis tools as a way to protect their current and previous software engineering investments.

Static source code analysis tools offer OEMs an additional means to quickly and objectively audit the work performed by distributed team members or third-party companies.

Complex, Distributed Engineering Teams Utilize Static Analysis

Time-to-market and cost considerations will continue to drive embedded systems manufacturers to lower-cost labor markets and third-party engineering service providers. Static source code analysis tools offer OEMs an additional means to quickly and objectively audit the work performed by distributed team members or third-party companies.

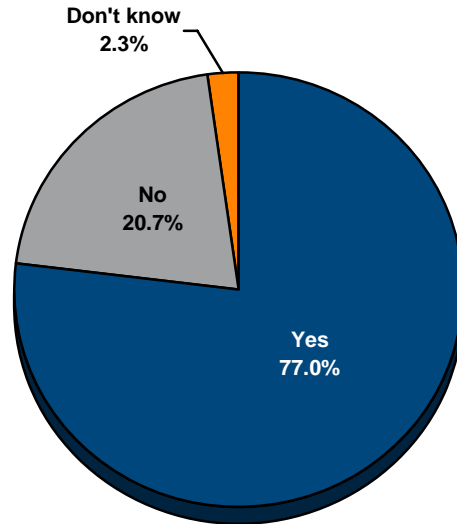
Engineering Tasks Outsourced to External Companies for the Current Project
(Percent of Respondents)



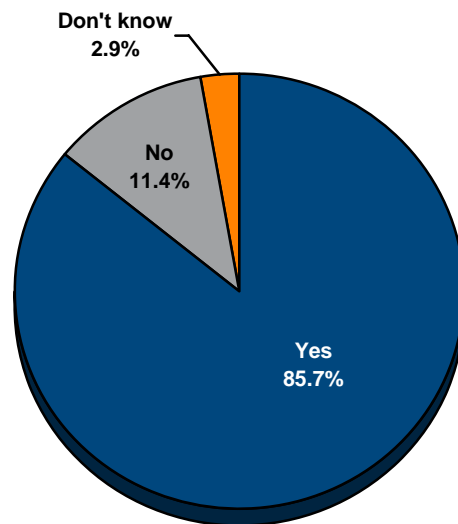
Note: Percentages sum to over 100% due to multiple responses.

Majority of Current Project's Engineering Team Located within Same Country as Respondent
(Percent of Respondents)

Using a Static Analysis Tool



Not Using a Static Analysis Tool



Surveyed static analysis tool users reported higher rates of geographic distribution within their internal projects teams as well as higher rates of outsourcing than the embedded software development community at large.

Signaling that engineers are using static source code analysis as a key tool within complex, distributed engineering teams, surveyed static analysis tool users reported higher rates of geographic distribution within their internal projects teams as well as higher rates of outsourcing than the embedded software development community at large. Moreover, the biggest difference between the outsourcing frequencies of the two populations was for software application/middleware development/test – arguably the most critical design task today to differentiate products and customer interaction – further signaling the utility of and potential need for static source code analysis.

Embedded systems manufacturers will continue to look toward resources abroad to lower costs and overhead and to improve corporate agility. As such, static analysis tools, coupled with other complementary lifecycle management tools, will play an increasingly critical role in OEMs' ability to mitigate the associated challenges and ensure product quality.

NEED FOR AUTOMATED TOOLS IN DEVELOPMENT OF MEDICAL DEVICES

As discussed above, the medical device industry, in particular, presents an intriguing paradigm within the embedded engineering landscape where many of the overarching trends affecting the broader embedded domain are magnified and accelerated as an unparalleled predicated rate of software code growth is juxtaposed with development teams that are smaller and have fewer software and test engineers (as a percent of team size). Moreover, the increasing software and system complexity within this sector is also often combined with additional safety- and/or security-critical requirements.

Despite these considerations, however, only 15% of engineers from the medical device industry reported using static analysis tools on their current project – one of the lowest adoption rates within the broader embedded domain. The exploding level of software content, connectivity, and sheer number of potential future device shipments within the medical device industry will necessitate that these manufacturers vigorously reevaluate their software development and testing practices.

Only 15% of engineers from the medical device industry reported using static analysis tools on their current project – one of the lowest adoption rates within the broader embedded domain.

Use of Static Analysis Tools on the Current Project for Software/System Engineering, by Vertical Market* (Percent of Respondents)						
Automotive/transportation	Consumer Electronics	Industrial Automation	Medical Devices	Military/aerospace	Mobile Phones	Telecom/datacom
29.6%	17.6%	12.0%	15.0%	17.5%	50.0%	19.4%

*Note: Certain vertical markets are excluded from this exhibit due to insufficient sample sizes.

As more functionality within medical devices becomes driven by embedded software – and with it the number of potential executable paths within the correspondingly larger code bases – an increasing proportion of medical device manufacturers’ duty of care will rest with ensuring proper software functionality. The FDA’s *Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices* describes numerous objectives that can be satisfied through the use of automated test tools. Dynamic testing, however, for which documentation is required by the FDA, is very useful for functional testing but cannot provide the same level of analysis for possible run-time errors within larger code bases as static analysis. Consequently, we expect that the adoption of static source code analysis tools to increase as a means to address and satisfy these objectives in those cases where other forms of automated testing are no longer adequate alone.⁵

If the other embedded devices sectors are any indication, this industry will need to drastically alter the amount of manpower and/or tool budget devoted to testing.

Static source code analysis is only part of the answer, but it is a solution that can be implemented by the software development team (as well as by QA/Testing teams) with the goal of improving development productivity, reducing backend testing costs, increasing quality, supporting documentation requirements for verification and validation and minimizing the potential for tort liabilities.

SUMMARY FINDINGS

In the midst of unprecedented economic uncertainty, the need to quickly deliver highly differentiated products to market has never been of greater priority and necessity. A missed market window can not only lead to reduced margins and market share, but can also seriously impair a company’s ability to successfully emerge from tough economic times.

Sacrifices in quality assurance, however, can be of an even greater detriment to a corporation’s long term health than delayed time to market. Static source code analysis will become increasingly valuable across application classes – from safety-critical devices to consumer electronics – as the cost of aftermarket device failures continues to increase in regards to tort liabilities, product recalls, and damage to brand image. Static analysis can offer organizations a means to mitigate this backend risk.

Static source code analysis will become increasingly valuable across application classes as the cost of aftermarket device failures continues to increase.

⁵ U.S. Food and Drug Administration, [Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices](#), May 11, 2005.

Static analysis tools will shift from a sound testing procedure to a cornerstone of embedded system development best practices, especially within those industries where code quality is of paramount importance.

The embedded system manufacturers who are embracing static analysis tools are delivering more software content, signaling the inclusion of more sophisticated features, functionality, and the potential for greater product differentiation and sales. While static analysis tools are by no means a silver bullet for the development of sophisticated embedded devices or overarching corporate success, our research indicates that those teams with more complex development projects are utilizing static analysis at a higher rate. Given the expected increase in code size across the embedded sector and no signs of a decrease in distributed teams, it's clear that there will be an increased need for these tools in the future.

Aside from providing embedded system manufacturers with the potential to control product quality and financial risk, VDC expects that static analysis tools will shift from a sound testing procedure to a cornerstone of embedded system development best practices, especially within those industries where code quality is of paramount importance and software content and functionality is growing at rates exceeding that of the overall market.

ABOUT VDC RESEARCH GROUP

VDC Research Group (VDC) is a technology market research and strategy consulting firm that advises clients in a number of technology markets including: Automatic Identification and Data Collection, Embedded Hardware and Systems, Embedded Software and Tools, Industrial Automation and Control, Mobile and Wireless, and Power Conversion and Control. Using rigorous primary research and analysis techniques, the firm helps its clients identify, plan for, and capitalize on current and emerging market opportunities. We strive to deliver exceptional value to our clients by leveraging the considerable technical, operational, educational and professional experience of our research and consulting staff. During our nearly four decades of ongoing operation, we have had the pleasure of serving most of the world's leading technology companies, many high-profile start-ups, and numerous blue-chip early and later stage investors. Our products and services consist of research reports, annual research programs, and custom research and consulting services. Founded in 1971, the firm is located in the Boston area. Please visit our Web site at www.vdcresearch.com to learn more.

VDC Research Group, Inc.

679 Worcester Road | Suite 2 | Natick, MA 01760

T: 508.653.9000 | F: 508.653.9836 | E: info@vdcresearch.com | W: www.vdcresearch.com

