VDC Research
Insights for the Connected World

Standardizing Safety & Security With Static Analysis

License to Distribute: RogueWave SOFTWARE

By André Girard, Senior Analyst, with Chris Rommel, Executive Vice President

INTRODUCTION

Getting Software Right Has Never Been So Important

The importance of efficiently developing secure, production-ready code has never been so critical, and the challenges of designing software on-budget and within tight timelines have never been greater. Software is an increasingly fundamental element of OEMs' business plans, where it increasingly delivers value to customers, generates revenue, provides differentiation, and protects corporate or customer records. Besides the important business considerations, software must more frequently manage system connectivity and growing portions of embedded device functionality. These system requirements magnify the potential negative impact of safety and security issues when bugs or vulnerabilities are not discovered or fixed. In safety-critical environments, these responsibilities mean that software failure can result in significant human injuries and even death.

Unfortunately, development teams are already stretched thin and frequently miss production deadlines. Now, as product functionality is increasingly dependent on software, the complexity of systems under design is escalating and adding further challenges. Without improving efficiency, developers are stuck, spending too much time and effort getting code through testing rather than adding new, differentiating features. Many organizations seemingly face a devil's bargain of needing to choose between producing applications that are secure and compliant or are completed on time. For embedded development teams to achieve quality and productivity goals simultaneously, they must both implement best practices and use proper tooling resources. To generate secure, production-ready software more rapidly, organizations should:

- standardize coding practices through adoption of software development standards,
- enforce secure development processes by adhering to functional-safety standards, and
- properly utilize static analysis tools to help developers find and fix vulnerabilities, bugs, and gaps in compliance as early as possible.

Background on VDC

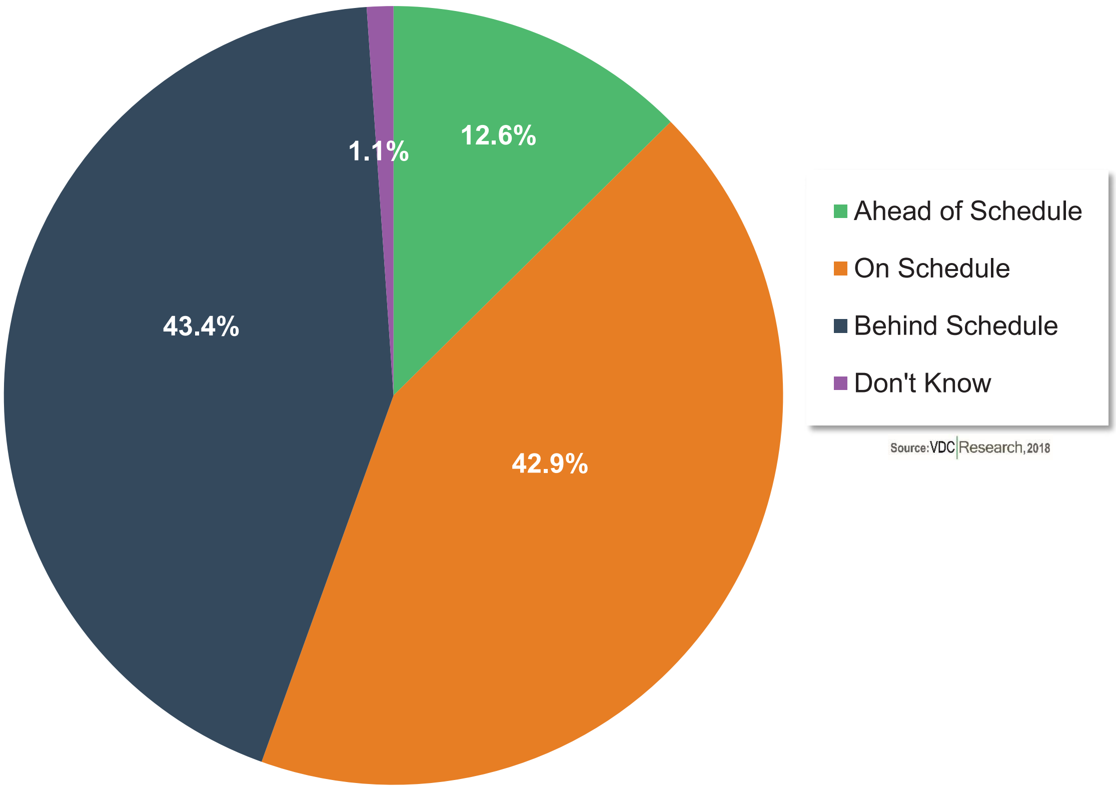
VDC has been covering the embedded systems market since 1996 and the use of lifecycle management solutions since 2000. Data supporting discussions in this paper is based on findings from VDC's most recent Software and System Development Survey. This survey captures the input from 539 embedded/IoT engineers and offers insight into leading business and technical trends impacting engineering organizations as well as the best practices to address them. The respondents are directly involved in software and systems development across a range of industries including automotive, aerospace and defense, telecom, medical, industrial automation, and consumer electronics, among others.

VDC Research
Insights for the Connected World

Development Challenges Highlight Need for Change

OEM development teams fail to meet project timelines with regrettably high frequency. In VDC's most recent Software and System Development Survey, 43.4% of embedded/IoT projects were reported as running behind schedule, with an average delay of 3.9 months. The regularity with which embedded development projects are completed late serves as an effective barometer to highlight practices needing improvement. On average, these delays cost embedded engineering organizations $115,000 each month that project completion is extended. But the total impact of these delays is far greater once opportunity costs are taken into consideration. These delays risk lost revenue from product sales or services from deployed devices, opportunities for developers to focus on creating important new features, and customers deciding to take their business to competitors.

Exhibit 1: Current Project's Schedule
(Percent of Respondents)

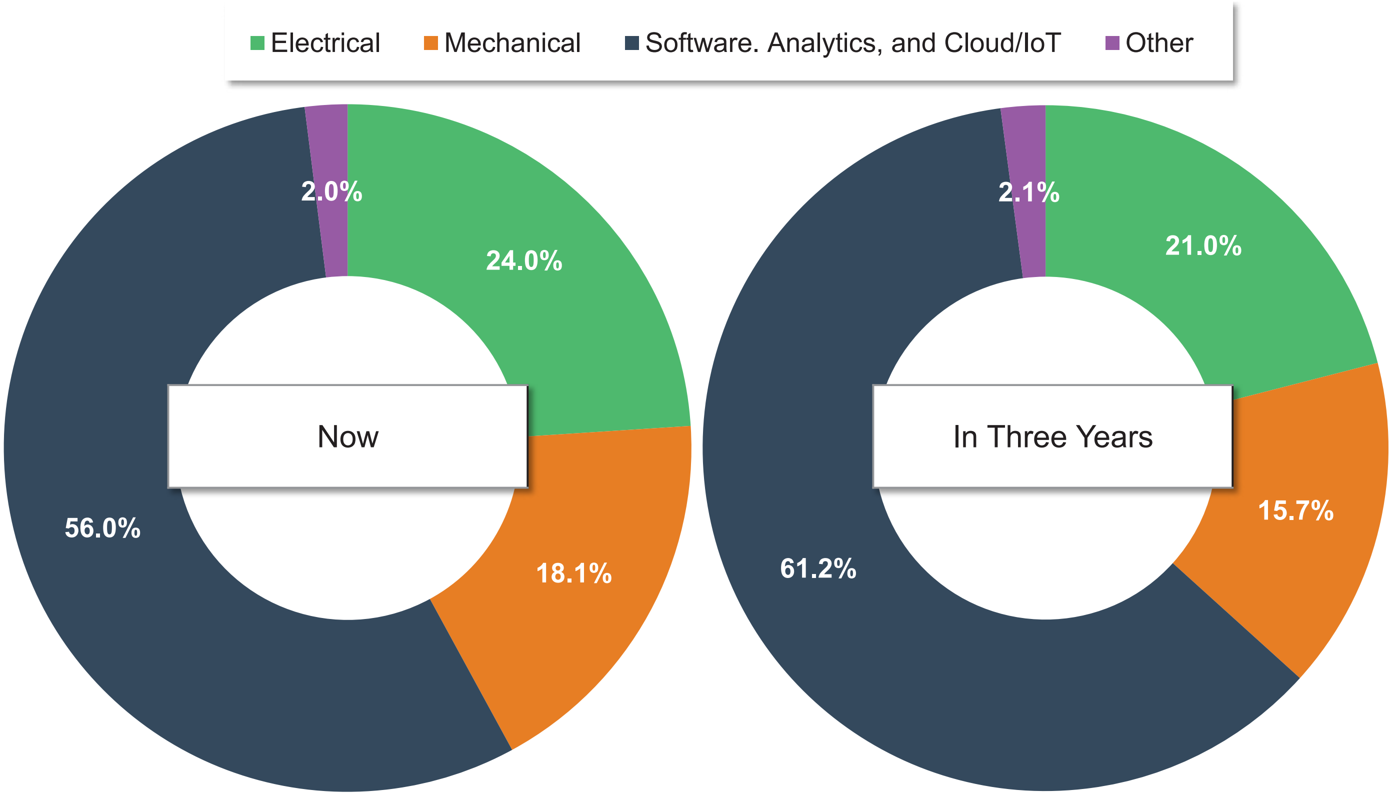


Source: VDC Research, 2018

Project schedule performance has been a long-standing issue across the embedded industries, in which the swift growth of system complexity continues to challenge the beneficial impacts of marginal operational improvements made by development teams. Unfortunately, many organizations fall behind because they are not investing in the resources and practices that can help maximize the efficiency of their development teams and the quality of the software they produce. The pace of change is unrelenting. Increasing demand for software-driven functionality and IoT connectivity is now adding even more complexity and technical obstacles — the two factors that are already the most frequently cited by engineers as contributors to delay — to system development. At the same time that these challenges grow, internal, customer, and industry or government compliance mandates threaten to add further confusion and strain for development teams.

Software — whether located on-device, in the cloud, or used for analytics — provides ever larger portions of embedded system functionality and value. As software drives more system capabilities, the proportion of development budgets consumed by developing these components is likewise rising. Surveyed embedded engineers indicate that the design of software-related components (on-device software, analytics, and cloud/IoT based software) already accounts for 56.0% of their current development budgets. These respondents expect software component expenses will grow to consume 61.2% of development budgets for similar projects in three years. Demand for IoT-related services and the advance of software-driven product innovation is encouraging OEMs to increase their software development expenditures. But continuing to scale software development spending in line with this functionality expansion is unsustainable, especially as the resulting technical challenges keep mounting. To meet market expectations for the on-going advance of embedded system capabilities within tight development timelines, organizations must find new ways to improve the efficiency of their engineering teams so that production of secure, high-quality software can increase faster than the related expenses.

Exhibit 2: Estimated Distribution of Development Costs on the Project (Average of Responses)

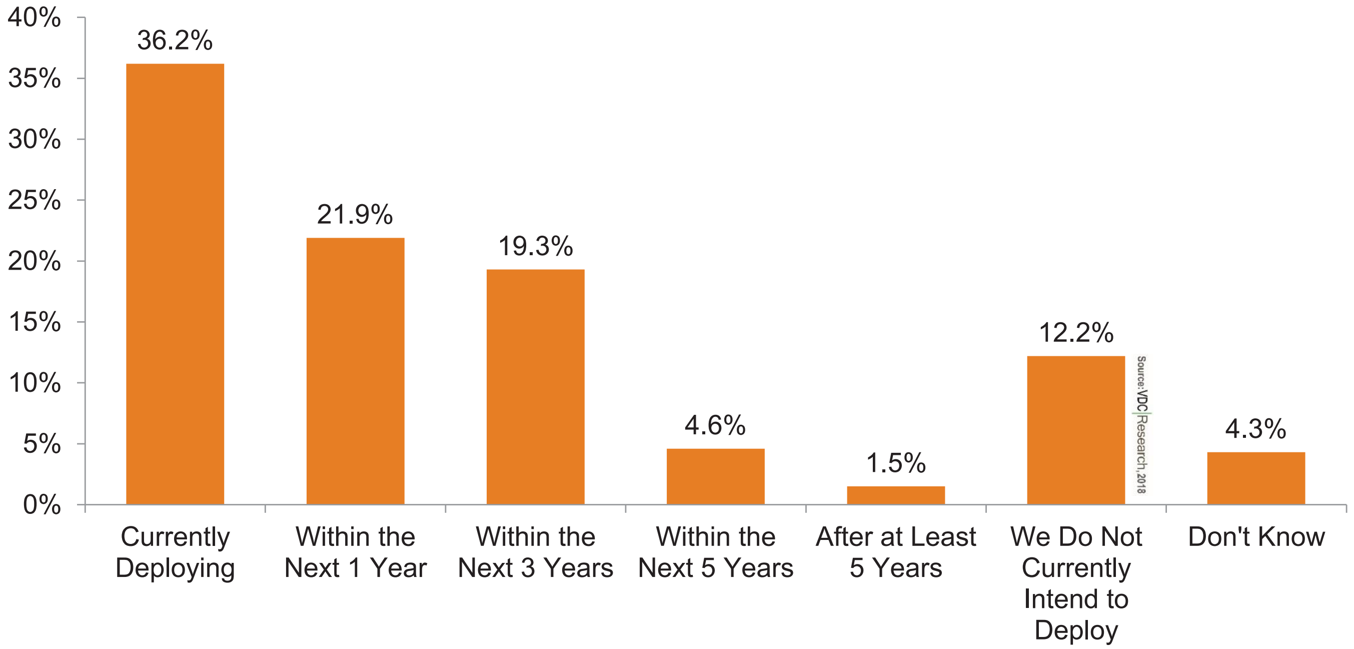


Source: VDC Research, 2018

IoT Drives More Innovation and Design Problems

The potential transformational impact of the IoT is fueling numerous project starts across the embedded industries. Already, 36.2% of engineers surveyed report their organization is incorporating IoT-capabilities or linking to applications through the internet. Another 41.2% expect to do so during the next three years. Engineering teams racing to design IoT systems within tight project deadlines must often rely on yet more new software to deliver these capabilities — adding even more volume and complexity to embedded code bases. This growth increases the risk that vulnerabilities will be introduced and highlights the need for organizations to improve software development techniques.

Exhibit 3: Deployment of IoT Capabilities and/or Applications into Organization's Products (Percent of Respondents)



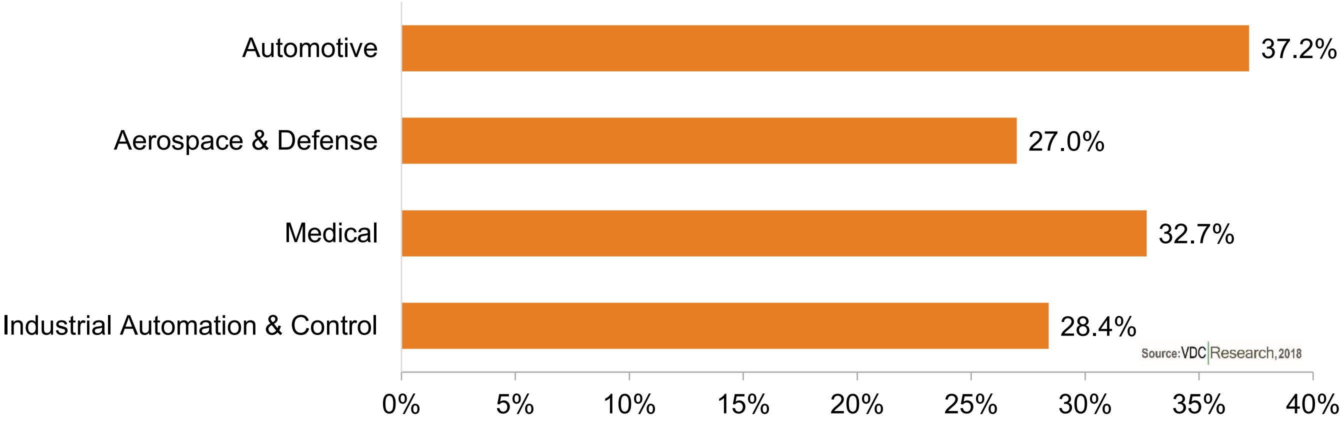
Beyond the need to increase the speed and efficiency of coding, the connectivity requirements of IoT projects bring the importance of security to the forefront. The networked architecture of these deployments greatly increases the potential impact of software vulnerabilities. Unfortunately, current engineering skillsets often lack knowledge of how to comprehensively address the security needs of IoT applications. Perhaps even more concerning than the skills gap is that — despite widespread awareness of the potential risks that software security vulnerabilities pose to corporate liability, brand reputation, and operator safety — proactive measures and investments to address security issues still fail to meet this growing need. For example, 22.9% of embedded engineers admit their current project included no actions taken in response to potential security risks, despite the fact that 92.1% believe security has some level of importance.

The potential consequences of failure in mission- or safety-critical applications adds further urgency to ensuring that software is designed correctly and vulnerabilities are avoided whenever possible. Several safety-critical software coding standards have been developed to help guide the design of safe software for these applications. Coding standards such as MISRA C/C++, CERT C/C++, HIC++, and JSF AV++ provide guidance on good programming style and practices for producing safe and reliable software that can be readily tested and maintained. Likewise, functional safety standards, such as ISO 26262 and IEC 61508, provide guidelines for system design to minimize system or device failures that could cause injury, harm, or death. Adhering to these coding and/or functional safety standards can help enforce valuable system development best practices, even outside of the safety-critical applications where compliance is often mandated.

Static Analysis for Coding Efficiency, Quality, and Security

Static analysis tools emerged as effective resources for improving software in safety-critical applications. The tools, which are able to conduct syntax parsing, data-flow analysis, symbolic logic checking, and code compliance review, can play a valuable role in improving the quality and security of the code bases that they produce. Static analysis tools can also provide a marked improvement in the efficiency of software design by finding bugs and violations of internal or external coding guidelines early in the development process, preventing the defects from impacting the work done by other engineers. Due to their effectiveness in rapidly finding gaps in compliance, the use of static analysis tools should be a key part of an organization's validation process for meeting certification testing requirements.

Exhibit 4: Static Analysis Testing Tool Use Rate in Safety-Critical Industries (Percent of Respondents)



Source: VDC Research, 2018

Despite the capabilities of modern static analysis, not enough embedded engineers are using the tools on their current project — even in industries where mission- and/or safety-critical software is common. A growing volume of projects in the market segments highlighted in Exhibit 4 must now demonstrate compliance with security, quality, process, or functional safety standards. Static analysis testing is beneficial in these situations, but is still not viewed as critical for use in some software components, such as those used in HVAC or other less vital applications. Nor is static analysis considered as important by all engineering roles or for all types of code. For example, engineers developing firmware are less likely to use the tools than teams focused on application-level software. However, until a significantly higher percentage of development teams utilize static analysis testing, particularly on projects facing compliance mandates, on-time schedule performance will remain poor across much of the embedded industries.

Challenges Are Driving Change Across Industries

The embedded/IoT market is a combination of industries with a wide range of customer expectations, technical requirements, and existing development practices. Despite these differences, most OEMs share similar challenges, such as the need to accelerate the pace of development, safely incorporate more third-party software, and prepare for an increasing volume of IoT deployments. The recommended strategy for addressing these challenges — adopting a more standardized approach to improving the safety and security of systems under development — is also shared. This approach should include organization-wide adherence to coding standards and the consistent use of static analysis tools.

Software failure from bugs, vulnerabilities, and security issues in safety-critical applications can have especially disastrous results. Companies developing systems for industries where safety-critical environments are common, such as automotive, aerospace, medical, and industrial automation, must employ especially rigorous coding practices. Many of these actions, however, can serve as best practices for organizations producing embedded software for applications outside of these industries.

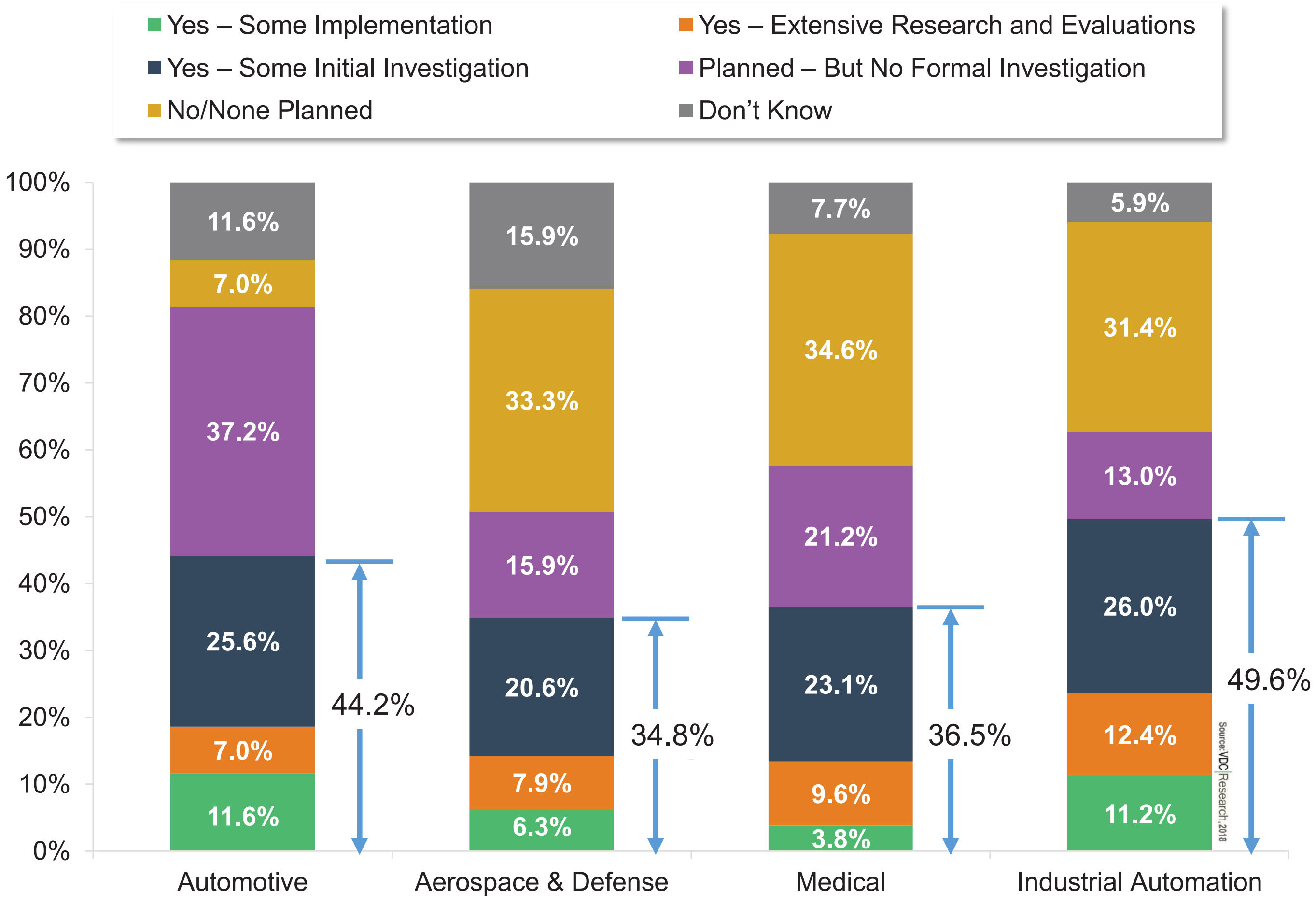
The specialized environments, technology needs, and requirements of the safety-critical markets call for a look at industry-specific considerations.

Automotive

Automotive OEMs face some of the most challenging engineering and business requirements in the embedded industries. Automotive systems are extremely complicated and are rapidly growing more complex as cars include more advanced driver-assistance systems and the industry advances towards autonomous driving. System functionality is increasingly dependent upon software that is highly integrated with mechanical and electrical components. The escalating complexity and the potential risk of software-related failures have encouraged industry and regulatory groups to impose extensive compliance standards.

Yet, all this resulting system complexity must be produced within tight profit margins and short development deadlines in comparison to industries, such as aerospace, that have similar levels of code base volume and complexity but much larger project timelines and budgets. These engineering challenges have forced OEMs to adopt rigorous development practices in order to survive. For example, over two-thirds of automotive engineers report that their current project either completely complies with MISRA C or that the standard is selectively enforced based on internal quality goals. The extent of integration across automotive engineering organizations is another example of process improvements made in response to development challenges. To help manage the system architecture complexity of automotive designs, the industry has the highest percentage of engineers working on projects which include an integration that extends across multiple engineering disciplines.

Exhibit 5: Implementation/Investigation of Multi-Engineering Domain (i.e. ALM, PLM, EDA) Integration (Percent of Respondents)

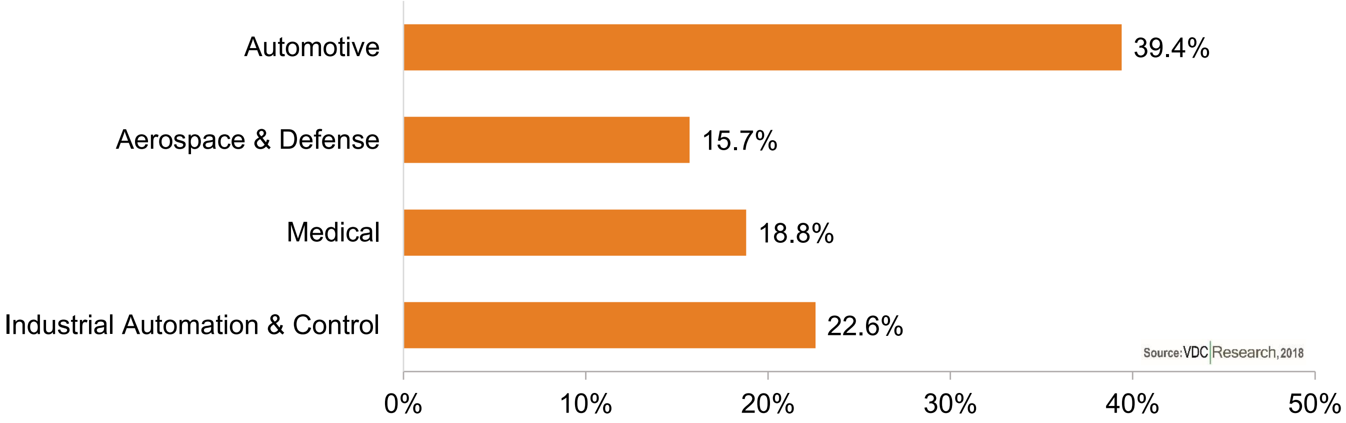


Automotive OEMs also have a comparatively long-established use of static analysis testing tools. However, historically much of this tool use has centered on code-checker types of functionality, rather than the deeper analysis of semantic issues in software and of security vulnerabilities. Additionally, the industry still relies too much on manual testing and other less capable tools.

Aerospace and Defense

Aerospace systems have long included some of the most complex embedded systems and required extremely large code bases. Developing avionics is very costly and system development projects in aerospace also face exceptionally rigorous compliance demands. For example, nearly any flying device must comply with the DO-178B/C process standard, whether a commercial or military aircraft, unmanned aerial vehicle (UAVs, aka "drones"), or even a missile. However, full compliance with software coding standards that can help instill disciplined, consistent coding practices trails other safety-critical industries and is an area in which OEMs should look to improve.

Exhibit 6: Full Compliance with a Formal Software Coding Standard on Current Project, by Safety-Critical Industry (Percent of Respondents)



Source: VDC Research, 2018

As the pace of technological change has accelerated, the lengthy development lifecycles of avionic systems have become more of a challenge. It is increasingly difficult to incorporate newer advances into aerospace systems because the aircraft design processes often began several years prior to the introduction of the technical innovation. Some major OEMs now focus more on systems integration, rather than manufacturing — outsourcing development of larger portions of the aerospace systems to suppliers. A key goal of this strategy is to shorten development lifecycles, thereby enabling the inclusion of newer technologies. However, this approach greatly increases the difficulty of coordinating the design, testing, and integration of components sourced from a broad supplier ecosystem.

Medical

The medical device manufacturing industry is much more fragmented than the automotive and aerospace markets. This fragmented component and manufacturing ecosystem makes comprehensive change challenging, but the regulatory environment and development practices in the industry are now evolving significantly. Despite the obvious safety-critical aspects of many medical devices, the industry's process and functional safety compliance guidelines were often framed as suggestions rather than requirements until recently. The US FDA and the European Commission now have issued numerous new medical guidance documents and ratified new regulations to address these issues. These new mandates place heavier compliance burdens on device manufacturers, to which they have rapidly responded by increasing the use of commercial development tools. The reported static analysis test tool use rate is strong in comparison to most other industries, at 32.7%. But many organizations are relatively recent adopters — use rates were only 28.2% in 2016.

Industrial Automation

While the industrial automation industry has traditionally been slow to change, it is now in the midst of an unprecedented transformation. The smart factory vision is ushering in what is commonly referred to as the fourth industrial revolution. In the smart factory framework, intelligent, cyber-physical manufacturing systems are connected over the internet to facilitate collaboration and data sharing. The complexity of new industrial systems is escalating as advanced technologies such as analytics, edge computing, and automation are more frequently incorporated. Many industrial automation OEMs are relatively well-equipped to manage the increasingly complex design requirements due to their long-standing use of product line engineering (PLM) and enterprise resource planning (ERP) tools — the vendors of which have evangelized the benefits of integration across engineering domains in the development of complex systems.

However, the rising demand for device intelligence and connectivity is forcing OEMs to reconsider existing security practices, and many organizations are not well prepared to respond to this changing market dynamic. The engineering teams in many industrial automation organizations will require training or an injection of new expertise. Due to the lengthy operating lifecycles for many industrial automation machines, many deployed systems were designed to operate in isolation or be connected only via a local area network. Security features were often not a consideration. The long lifespan of equipment adds additional difficulties, in that most deployments are necessarily a brownfield environment with a mix of old and new devices. Addressing security will be a challenging process that requires balancing integration and partitioning as well as linking old and new technologies.

Recommendations for Improving Software Safety and Security

The urgency of improving the safety and security of software being developed is most acute for organizations developing safety-critical applications. However, the best practices and resources for helping to meet the needs of today and prepare for the challenges of tomorrow can be universally applied across all industry sectors.

1. Follow Best Practices for Designing Quality Applications

- **Increase the focus on quality to improve operational efficiency:** Employing lengthy testing cycles to find and eliminate bugs before software release is expensive and time consuming. To achieve the coding efficiency necessary to avoid budget overruns and project delays, increase the focus on preventing the introduction of bugs and finding them in the earliest stages of development.
- **Invest in tooling and training:** OEMs need to increase investment in commercial development tools and training to ensure engineering teams have the resources and expertise to build high quality systems.
- **Re-evaluate product selection criteria:** Tooling selection must involve an evaluation of the usability of a tool and how well it is fit for purpose to ensure it will be used consistently and address the organizations' needs. The total cost of ownership for free or low-cost tooling can far exceed that of premium tools if poor usability limits their operation or issues like false positives or negatives negate their usefulness.

2. Arm Your Organization with the Best Static Analysis Tools

By finding coding violations at the earliest point possible, static analysis testing is an important part of initiatives to improve the quality and security of software and increase the efficiency of engineers. However, not all static analysis tools will provide similar performance, nor will inconsistent use provide the same benefits. Companies must select the right tool and ensure it is operated correctly.

The right static analysis tools will be those that, besides finding semantic and syntax errors, can also help detect security vulnerabilities and achieve standards compliance. Static analysis tools can be certified for usage in standards-compliant development processes. The evaluation process must include verification that the tool is certified to the standards an organization faces to avoid the time and significant expense of independently going through the tool certification process. Additionally, companies should select tool suppliers that actively support and update their platforms to best prepare the organization for newly emerging threats and compliance standard revisions.

To maximize the benefits of static analysis, the same tool should be used across the development team so rules are applied and discoveries identified consistently. Static analysis testing should begin as early as possible. Highly capable static analysis tools, such as those from Rogue Wave Software, enable operation on the developers' desktops. They should also be used as part of the code check-in process and throughout the build as part of a continuous integration process to prevent errors from being inadvertently introduced.

3. Follow Coding and Functional-Safety Standards for Establishing Optimum Processes and Results

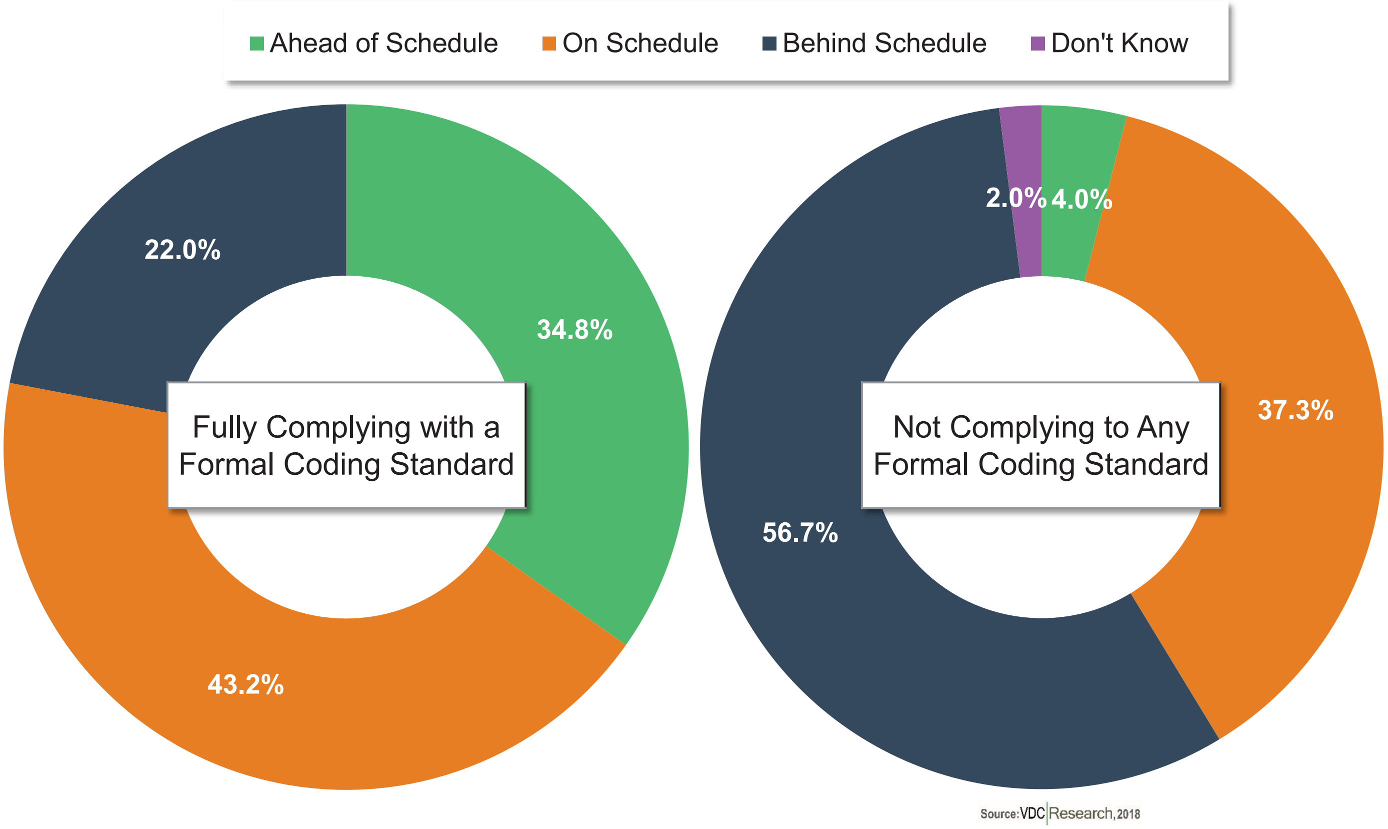
Recommendations or mandates to follow functional safety and process standards are growing more common and rigid. Organizations need to recognize the importance and benefits of standards compliance and start preparing now for the demands of compliance. Emphasizing the benefits of getting all of the development teams following the same procedures will help encourage engineers to buy in.

Following the directions laid out in process or functional-safety standards provides organizations with guidelines that encourage or enforce best practices for software development. They help to prevent coding violations and avoid many common software vulnerabilities. In some cases, adherence to process or functional-safety standards will be prescribed by customers, supply-chain partners, or regulatory agencies.

Even when it is not required, adherence to coding standards can instill disciplined coding practices that help teams design production-ready software more rapidly. The collaborative peer-review processes organizations often put in place to support standards compliance results in fewer errors being introduced into the codebase. This, in turn, reduces the workload on their quality assurance (QA) and test engineer teams because fewer mitigating issues are discovered and need to be fixed later in the development lifecycle.

> *"Software projects in full compliance with coding standards are over eight times as likely to be ahead of schedule in comparison to projects not applying a standard"*

Exhibit 7: Current Project Schedule Performance, by Coding Standard Compliance (Percent of Respondents)



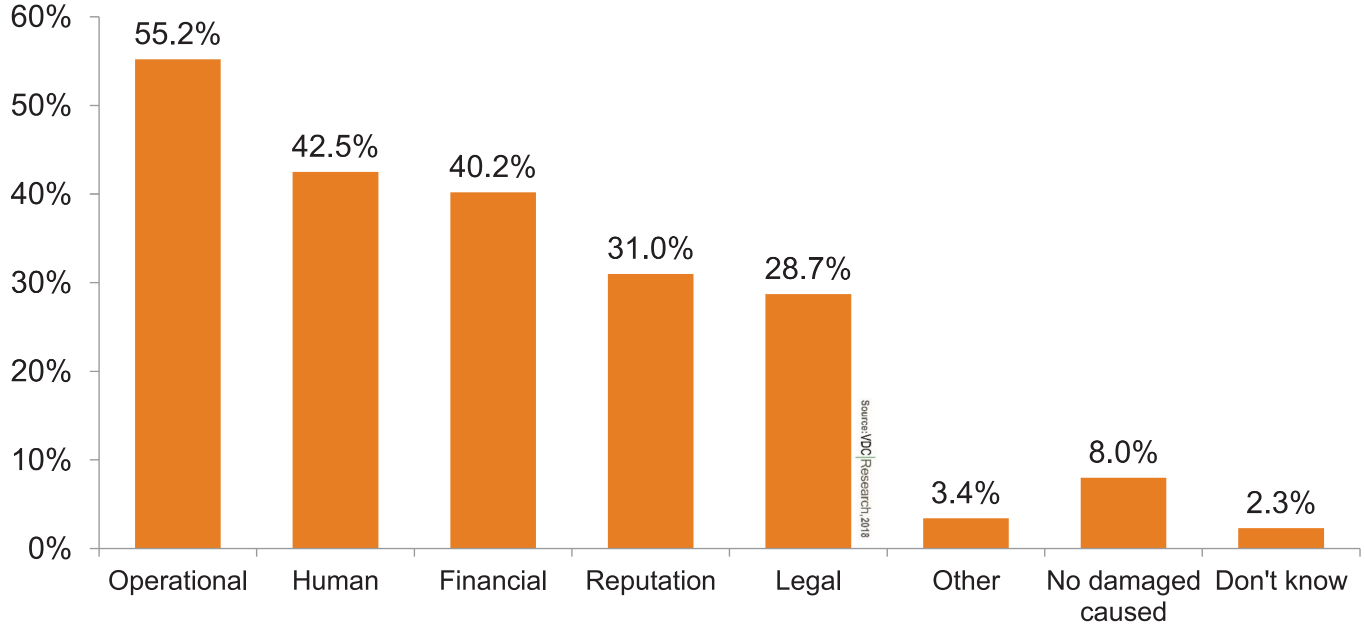
Source: VDC Research, 2018

Recent survey results demonstrate the significant improvement in schedule performance on projects where software was produced in full compliance with a coding standard. In comparison to projects where no coding standards were applied, fully-compliant projects were more than eight times as likely to be ahead of schedule and less than half as likely to be behind schedule.

4. Start Security Mitigation in the Earliest Planning Stages

Many organizations apply security measures late in the development lifecycle due to uncertainty about the best practices to take for minimizing risk. Other teams apply inadequate half measures that only address security concerns for some system components in some development stages. But the risks are too high and consequences of vulnerabilities are far too severe to take no action or only inadequate, partial steps. Organizations that encounter security vulnerabilities can suffer significant damages in several domains, as shown in Exhibit 8. Many engineers report multiple areas of impact — and this is only taking into consideration issues that the organization has discovered.

Exhibit 8: Implications of Security Vulnerabilities
(Percent of Respondents)



Source: VDC Research, 2018

To establish the best protection from potential impacts, security must be improved at every development phase, not just bolted on at the end of the development lifecycle or upon deployment. The most effective security mitigation strategies must begin in the early stages during planning, and continue through the full system lifecycle. System architectural design decisions should be made with the goal of minimizing attack vectors. Likewise, security considerations should help guide the selection of system components such as a security-hardened OS or processors that enable an encryption solution to directly tie in.

5. Integrate Across the Engineering Organization

Engineering organizations designing the newest generation of embedded systems need high levels of coordination and communication, not only across the software development teams, but also between engineering groups focused on other domains. Improving the integration across engineering disciplines facilitates channels of communication that can help increase development speed and improve accuracy. Cross-engineering domain integration also prepares the engineering organization to design the system of system architectures of IoT deployments.

As the functionality of electronics, hardware, and software components continue to grow more dependent upon each other, architectural decisions and change management processes increasingly necessitate coordination across all the engineering teams involved in a system's design. Tight time-to-market demands, increasing system complexity, and rapidly evolving requirements only serve to further raise the importance of information sharing and coordination between the formerly independent engineering domains.

ABOUT THE AUTHORS

André Girard

André Girard brings valuable perspective to the market research and consulting of the IoT & Embedded Technology team, having previously covered connected devices for both the Telecom and Embedded Hardware practices at VDC. His primary areas of expertise include lifecycle management solutions, Agile development, and cross-domain engineering integration. André's IoT technology background includes opportunity sizing and forecasting, market and technology assessments, competitive analysis, strategic marketing assistance, and M&A due diligence support. He also gained important experience as an independent consultant covering telecommunications and the smart grid. André holds a B.A. (magna cum laude) from MCLA.

Contact André:
agirard@vdcresearch.com

Chris Rommel

Chris Rommel is responsible for syndicated research and consulting engagements focused on development and deployment solutions for intelligent systems. He has helped a wide variety of clients respond to and capitalize on the leading trends impacting next-generation device markets, such as security, the Internet of Things, and M2M connectivity, as well as the growing need for system-level lifecycle management solutions. Chris has also led a range of proprietary consulting projects, including competitive analyses, strategic marketing initiative support, ecosystem development strategies, and vertical market opportunity assessments. Chris holds a B.A. in Business Economics and a B.A. in Public and Private Sector Organization from Brown University.

Contact Chris:
crommel@vdcresearch.com

ABOUT VDC RESEARCH

VDC Research
Insights for the Connected World

Founded in 1971, VDC Research provides in-depth insights to technology vendors, end users, and investors across the globe. As a market research and consulting firm, VDC's coverage of AutoID, enterprise mobility, industrial automation, and IoT and embedded technologies is among the most advanced in the industry, helping our clients make critical decisions with confidence. Offering syndicated reports and custom consultation, our methodologies consistently provide accurate forecasts and unmatched thought leadership for deeply technical markets.

Located in Natick, Massachusetts, VDC prides itself on its close personal relationships with clients, delivering an attention to detail and a unique perspective that is second to none.

© VDC Research Group, Inc. | P 508-653-9000 | info@vdcresearch.com | www.vdcresearch.com