2017

Embedded Systems Safety & Security Survey

20251 Century Boulevard Ste. 330
Germantown, Maryland 20874

Telephone: +1 (866) 65-EMBED
www.barrgroup.com

Table of Contents

Table of Figures

Executive Summary

In 2017, Barr Group conducted a survey of the embedded systems industry. A total of 1,726 survey responses from active, professional system designers were received from engineers with an average of 16.7 years of paid experience. Respondents were employed in companies of all sizes, about equally within and outside the United States, and across a broad range of vertical markets.

After carefully analyzing the response data, Barr Group's key findings regarding the current state of safety and security practices of embedded systems designers are:

- **There is a large opportunity to easily improve the safety of embedded systems by more broadly using well-known software development best practices.**
- **Safety practices are not clearly better in the automotive industry than in the medical device industry—even though many more lives are at risk with automotive failures.**
- **Broader use of software development best practices is also an opportunity to better secure the vast numbers of Internet-connected devices to come.**
- **Designers of a remarkably large number of potentially dangerous, Internet-connected embedded systems are ignoring security altogether.**
- **Because the range of architectures and applications is large, there will never be a one-size-fits-all solution to the problem of securing embedded systems.**

In brief, there are potentially deadly embedded systems that are not designed with appropriate levels of care as well as systems that could be more secure. There is, thus, much work to be done in the embedded systems design community to achieve a safer and more secure world. Fortunately, a lot of what needs to be done is well understood and easy to implement; what appears to be lacking is motivation.

About Barr Group

Founded by internationally known experts in the design of safe and secure embedded systems, Barr Group is an independent provider of world-class consulting, training, and product design services. From pacemakers to cars, The Embedded Systems Experts[1] at Barr Group help make the computers inside everything safer, more reliable, and more secure.

As part of its mission to improve the whole industry, Barr Group conducts the *Embedded Systems Safety & Security Survey*™. With the highest response rate of any survey in the industry, this annual survey of the engineers who are on the front lines in the design of products that will soon come to market in a range of industries provides valuable industry insight into design trends and development practices.

Consistent with this mission, the Barr Group website is replete with how-to technical articles and other free resources for embedded systems designers. The company also produces free quarterly webinars on various topics, which are widely attended live and also made available for later playback from its website.

In terms of its business, Barr Group specializes in providing unbiased embedded systems process and (re)architecture consulting services to directors of engineering, technical managers, and their teams. Many types of engagements are possible and each consultant is a senior engineering expert who communicates clearly and effectively in writing and in person. More information about Barr Group's consulting services can be found at http://www.barrgroup.com.

---

[1] Barr Group, the Barr Group logo, and The Embedded Systems Experts are registered trademarks.

Barr Group also trains engineers and its world-class courses are designed to strengthen critical programming and engineering skills for embedded system design teams across all industries. Through these courses—such as the four-day, hands-on *Embedded Software Boot Camp*® and *Embedded Security Boot Camp*®—engineers learn the important development skills needed to efficiently design safer, more reliable, and more secure applications. Barr Group offers public training in North America and Europe, as well as private and custom training courses all over the world.

Because Barr Group's engineers are independent-minded experts capable of researching tough subjects and adept at explaining complex technical topics in everyday language, Barr Group consultants have often been called upon to testify as expert witnesses in patent infringement, intellectual property, product liability, and other technical legal disputes. Notable expert testimony from Barr Group experts has related to the security of satellite communications systems and smartcards, smartphone industry patents, software copyrights in video games and multi-function printers, as well as the Toyota unintended acceleration personal injury litigation.

Finally, Barr Group's *Embedded C Coding Standard*™ has been adopted and adapted by thousands of embedded programmers and teams. The coding standard was created to help developers minimize bugs in firmware by focusing on practical rules that keep bugs out—while also improving the maintainability and portability of C/C++ code. Published as a print and electronic book as well as on the Barr Group website and fully compatible with MISRA's "Guidelines for the Use of the C Language in Critical Systems" subset of the language, the *Embedded C Coding Standard* details a set of guiding principles, naming conventions, and stylistic rules for the use of data types, functions, preprocessor macros, variables and much more. The individual rules that have been demonstrated to reduce or eliminate certain types of bugs are highlighted.

Background and Methodology

Barr Group's annual *Embedded Systems Safety & Security Survey* is a web browser-based online survey. The survey is designed to be easy to answer and usually requires no more than about 5 minutes to complete. This year's survey consisted of 34 multiple-choice questions and was hosted at SurveyMonkey.com.[2]

Outreach and Response

The survey was open from January 10, 2017 until February 3, 2017. After final editing and internal testing of the skip-logic, a "beta test" was performed on January 10 by inviting several hundred attendees of a Barr Group webinar to participate via a link at the end. No problems were found with the survey during the beta test, which thus provided an initial batch of about 100 completed survey responses.

We subsequently leveraged the Barr Group mailing list of over 30,000 addresses in combination with other mailing lists for embedded systems designers to send more than 180,000 total emails containing invitations to the survey. In addition, we announced the survey and provided links on our website at BarrGroup.com, in social media (specifically, LinkedIn, Twitter, and Facebook), and on EETimes.com.

As an incentive to participate as well as a thank you for their valuable time, those who completed the survey and also provided their email address were each given a chance to win one of two Saleae Logic-8 USB logic analyzers (retail cost $219) or one of three Amazon.com gift cards ($25 value). It was not required to provide an email.

---

[2] Each of the questions and its possible answer choices is provided for reference in Appendix A. The full set of response data collected from the qualified respondents is provided in Appendix B.

This year's survey results are drawn from 2,022 completed survey responses.[3] This number is lower than the number of people who took the survey, as sometimes the respondent to a web-based survey will begin to answer a survey and subsequently become distracted or otherwise abandon the survey prior to completion. Incomplete responses were periodically removed from the dataset and the number of incomplete responses was not tracked.

Also excluded from the 2,022 completed survey responses were several dozen "duplicate" responses, which were apparently submitted by the same person.[4] After the survey was closed, we sought duplicates in several ways (e.g., by searching the data for duplicate email addresses). In each such case, only the first (by date and time) complete response was retained.[5]

Some of the 2,022 complete survey responses were not from professional embedded systems design engineers.[6] For example, some were responses from graduate students interested in embedded systems, professors, or others (such as company executives) who are not directly involved in the design of any specific product.[7]

---

[3] An individual survey response was considered completed if all of the "required to answer" questions presented to that person (based on skip-logic) were answered.

[4] Although the survey platform restricted multiple responses from the same IP address, some people may have taken the survey more than once in hopes of winning a prize or simply because they were sent multiple invitations through multiple mailing lists.

[5] Brief scans of suspected pairs revealed that the answers to various questions were typically identical.

[6] Though the majority of the non-qualified respondents may have tangential connections to the embedded systems industry, analyzing this data would have made the overall findings less accurate.

[7] Nevertheless, all who provided an email address were given an equal chance in the prize drawings.

To improve the quality of the data and analysis, skip logic embedded in the flow of survey questions was used to narrow the group that answered the more detailed questions. For example, when a respondent answered the question "*How much professional experience (paid work, not counting academic work) do you have in the field of embedded systems design*?" with "*I have no professional experience in embedded systems design*." only a few demographic questions were presented.

This technique reduced the qualified dataset in the following manner:

- 147 respondents had no paid years of design experience;
- 80 respondents were not directly involved in product design; and
- 69 respondents were unable to adequately identify a current project[8].

The remaining set of 1,726 completed survey responses is believed to be entirely from paid/professional embedded systems designers who are actively working on an identifiable design project. The data and analysis presented in this report is drawn only from this subset.

[8] Or were designing a tool to assist embedded systems designers in their work rather than an end product that is itself an embedded system.

Statistical Significance

With its sample size of 1,726, this survey is mathematically calculated to have a confidence interval of +/- 2.4% at a confidence level of 95%.[9] More simply put, the true percentage across all professional embedded systems designers is 95% likely to lie within +/-2.4% of the measured sample. For example, if 60% of those surveyed have adopted a coding standard, the actual percentage is almost certainly between 57.6% and 62.4%.

Note, however, that the surveyed group of 1,726 designers may not qualify as a randomly-selected group of the overall universe of professional embedded systems designers. That is, there are probably biases inherent in the methods of the invitation process, such as using English to communicate as well as certain mailing lists. Likewise, there may be certain subgroups within those invited who are more likely to open industry emails and/or participate in online surveys.

In some sections, the survey data analyzed in this report is with respect to a subset of the responses. For example, two small but important subsets are:

- A subset of 475 who are designing potentially dangerous systems, with a confidence interval of +/-4.5%.
- A subset of 226 who are designing potentially dangerous systems that will also be Internet-connected, with a confidence interval of +/-6.5%.

[9] See, e.g., http://www.surveysystem.com/sscalc.htm

Respondent Demographics

Before presenting the detailed analysis of embedded system development processes and architectures, it is worthwhile to consider respondent demographics.

Where They Live

We sought and received survey participation from English-speaking embedded systems designers wherever they were in the world.[10] The worldwide distribution of qualified survey respondents was as shown in Figure 1.

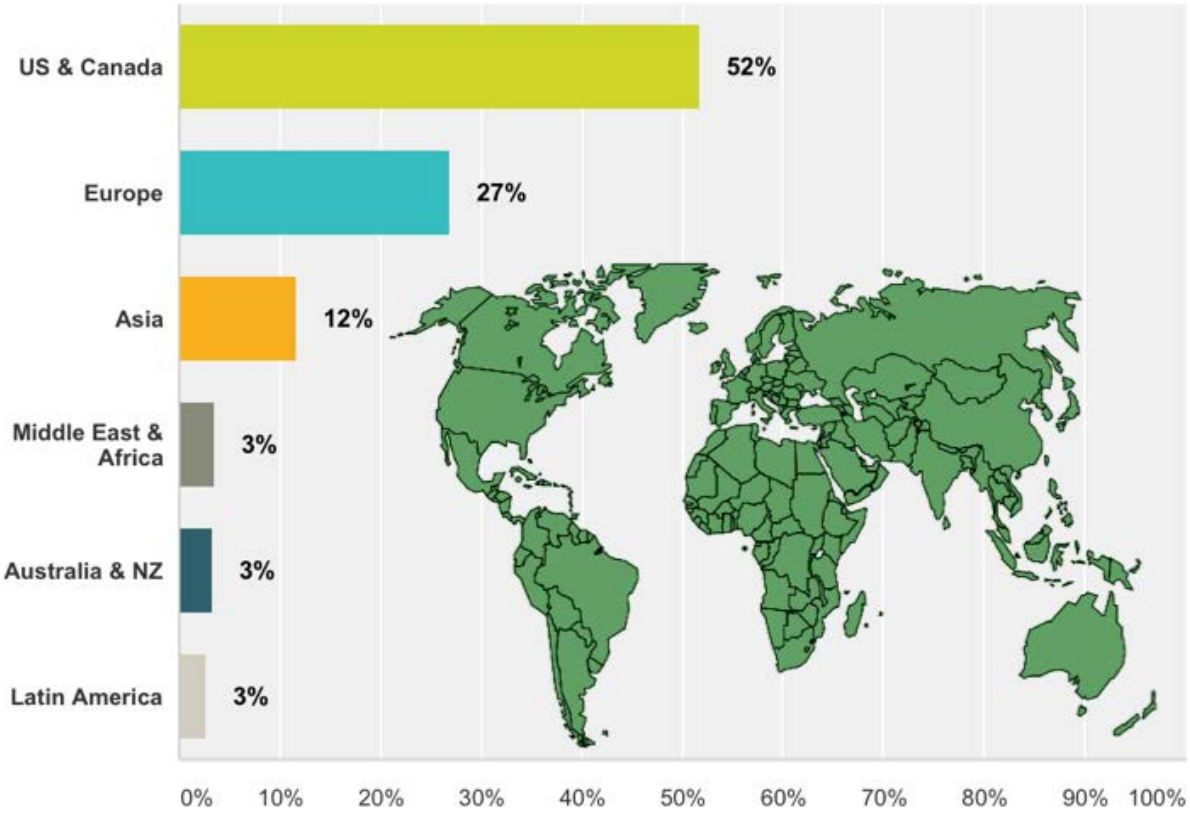


Figure 1. Worldwide Distribution of Surveyed Embedded Systems Designers

[10] Design engineers who don't speak English and/or don't subscribe to industry news in English were likely missed.

Within the United States, the regional distribution of survey respondents was as shown in Figure 2. (Respondents picked their individual state from a drop-down list of 52 options. We subsequently combined the state data into the nationally-recognized economic regions shown.)

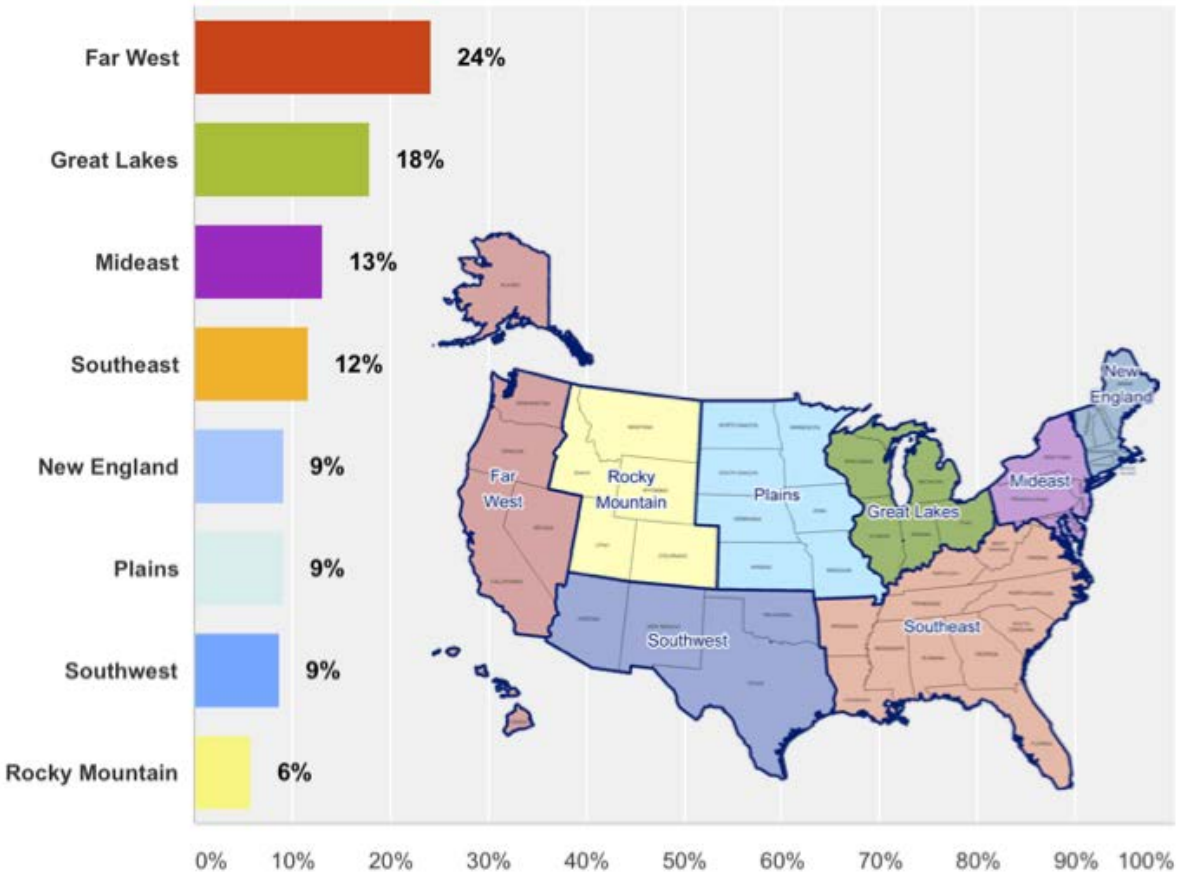


Figure 2. Regional Distribution of U.S.-Based Embedded Systems Designers

Roughly consistent with their relative population sizes, survey responses from Canada were approximately one-tenth of the total from the U.S. & Canada.

What They've Done

Although the largest percentage (36%) of qualified survey respondents were still in a group in the first decade of paid embedded systems design experience, the average respondent had a long design career spanning already nearly 17 years.[11] As also shown in Figure 3, the average number of years of paid experience was much higher in the United States (20 years) than in Europe (14) or Asia (11).

Remarkably, the experience distribution in the U.S. is effectively flat, with a slight bias up toward 20+ years of design experience. This combined with the high average likely reflects both the aging of embedded systems designers and that younger engineers and programmers are entering other industries.

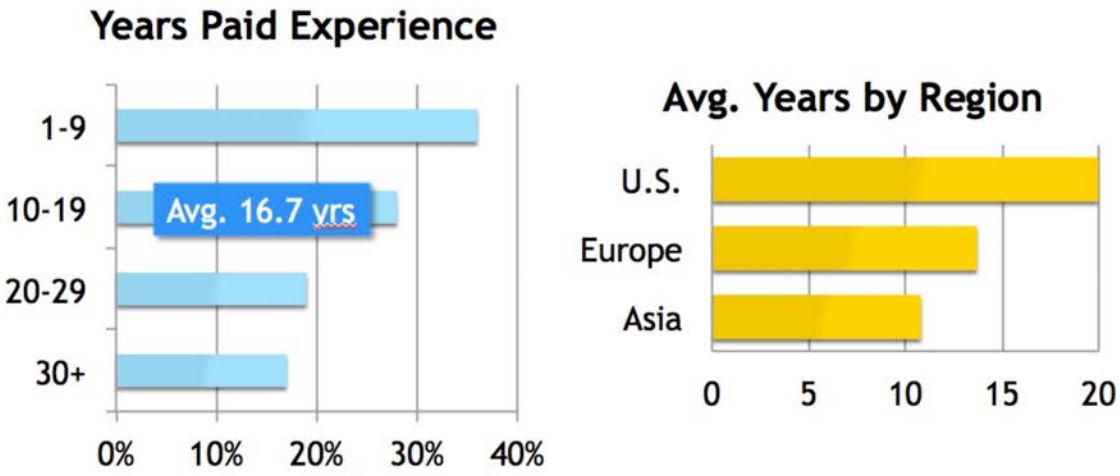


Figure 3. Years of Professional Embedded Systems Design Experience

[11] Averages were computed as the weighted average of the midpoints of each answer group (i.e., 5 years was used for the 1-9 group, 15 years for 10-19, 25 for 20-29, and 35 for 30+).

Where They Work

Embedded systems are products destined for a wide-range of vertical markets. Some will become subsystems in a complex product, such as an automobile or a fighter jet; some may be one-a-kind and travel to distant worlds. Others are simple standalone children's toys. Respondents to this year's survey indicated that their current projects were targeting a diverse range of industries, as shown in Figure 4.

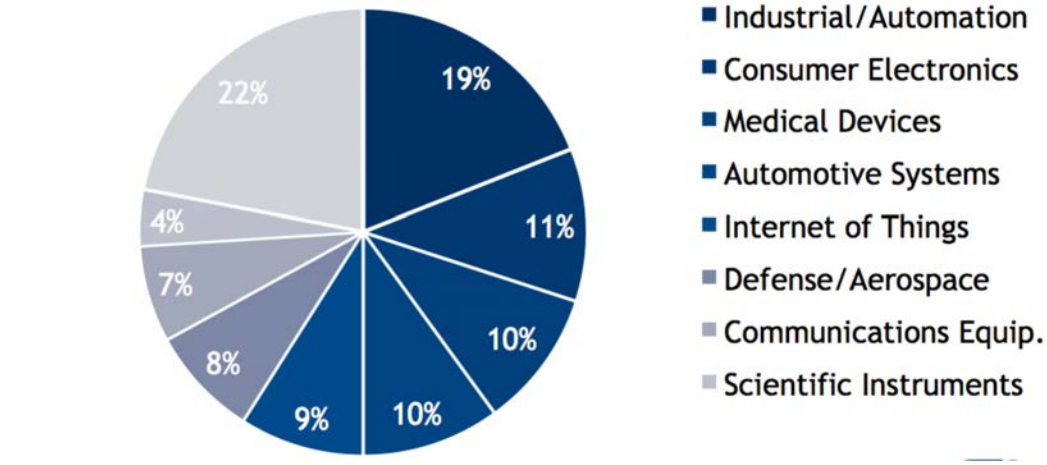


Figure 4. Vertical Markets Currently Targeted by Survey Respondents

Figure 5 presents data concerning the size of the organizations studied. The survey results represent a broad sample of the design practices of companies in a range of sizes, from the tiniest startups to the very largest multi-nationals.

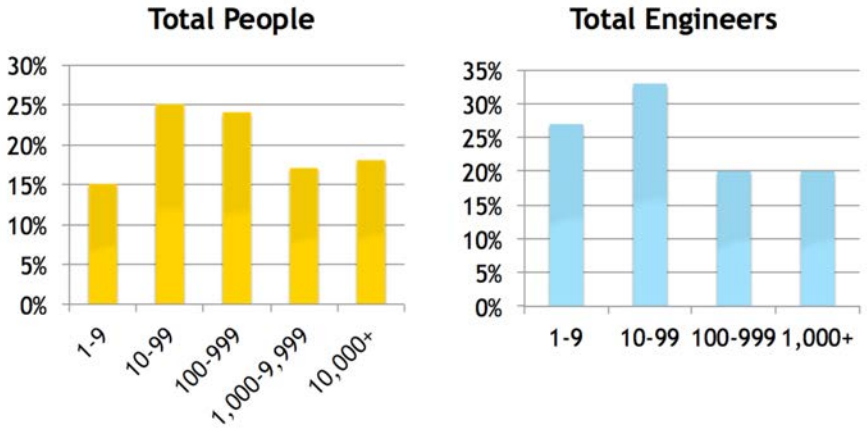


Figure 5. Sizes of the Organizations from Which Respondents Participated

A broad range of companies was represented. A sampling of the organizations from which embedded systems designers participated is shown in Figure 6. This is merely a sample and does not include the names of numerous other companies and organizations.

Agilent * Alcatel * Ametek * Apple * Aquatron Robotics
Battelle * Bayer Healthcare * Beta Bionics * Borg Warner * Bosch
Calsense * Carrier * Cisco * Continental Automotive * Cruzio
Daimler * Dolby * Eaton * Echostar * Fluke * Ford * Fresenius Medical
Garmin * Goodyear * Graco * Grundfos * Harman * Harris * Honeywell
Hughes * IBM * Intel * JHU-APL * John Deere * Keysight * L-3 Communications
Landis+Gyr * Lenovo * Lincoln Electric * Lockheed Martin * Lutron
MED-EL * Medtronic * Mitre * Motorola * NCR * NDI Medical * Netapp
Northrup Grumman * Omron * Orthoscan * Overhead Door * Philips * Phytec
Qualcomm * Renesas * Rockwell * Rockwell Collins * Schneider Electric
Schonstedt * Sciex * Seagate * Shlumberger * SnapOn * Spirent * Stryker
Teledyne * Thales * Thermo Fisher * Texas Instruments * Toshiba
Tyco * Visteon * Wavetronix * Whirlpool * Xerox * Zebra

Figure 6. Some of the Many Organizations Represented in This Year's Survey

What They Do

By definition, the design of an embedded system involves the design of both electronics (i.e., "hardware") and associated embedded software (a.k.a., "firmware"). On small projects, a single engineer may do both. In larger projects, a team of hardware designers, firmware developers, and testers work together. Typically, the software subgroup is the largest and includes some hardware-software dualists.

Figure 7 shows the distribution of the sizes of software-development teams, during the period of peak effort. Importantly, over two-thirds of software teams never have more than 4 people and only about 15% ever have 10 or more. Figure 7 also shows the primary roles of those who responded to the survey.[12] The largest group (around 50%) primarily develops software. The smallest group develops only hardware, though the second largest group does both. The rest are managers and system-level architects.

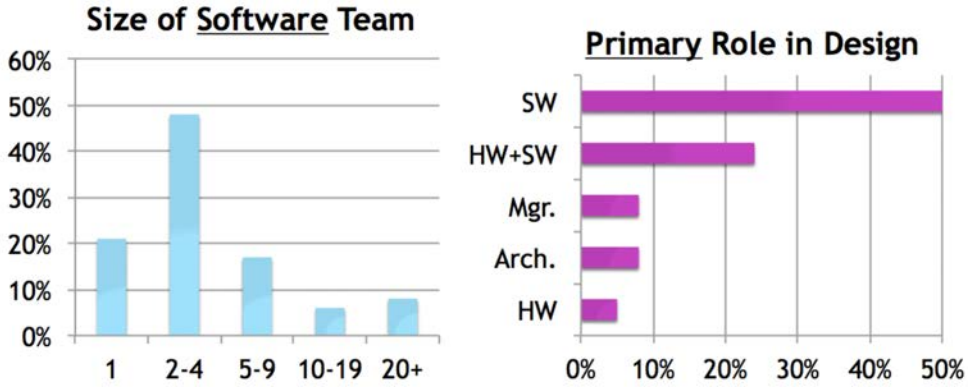


Figure 7. Peak Software Development Team Size and Respondent Primary Roles

[12] As mentioned above, we disqualified survey takers who indicated they worked in academia and those in executive management roles. And because we were only interested in the design of the hardware and software, we also disqualified those employed primarily in testing roles.

Industry Snapshot

Nearly all of the survey questions were asked in the context of a "*single embedded systems design project you are currently involved with.*" Reminders of this context were place at the top of each relevant page of the survey. As well, the phrase "*your current project*" was made part of the phrasing of questions where necessary to aid clarity.[13]

Processors

Over the last decade the number of processors (including microcontrollers and cores) in a typical embedded system has grown substantially, as can be seen in Figure 8. This year, only a third of new designs had a single processor. At the other extreme, a quarter had 4 or more processors. The largest group had either 2 or 3 processors.

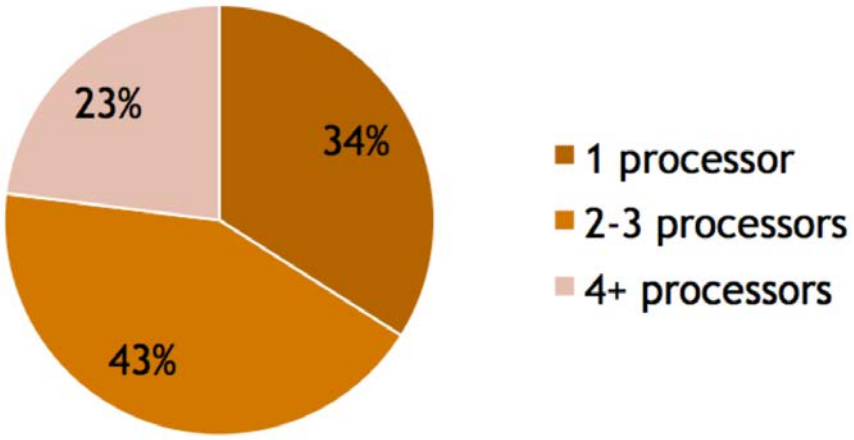


Figure 8. Number of Processors in Current Embedded Systems Designs

[13] Because—even with reminders like these—humans are not always reliable/consistent, we took the added step of disqualifying a few dozen of the respondents who answered "I don't know" to one or more of a set of base-lining questions.

Operating Systems

In our experience, it is most commonly the case in multi-CPU designs that there is one primary processor that may run some type of commercial or open-source operating system and this is then surrounded by either cores or microcontrollers that are much more likely to have no formal operating system. Rather than try to get at all of these details, which would be difficult in a multiple-choice survey, we asked very directly about the type of operating system on the "primary processor". Figure 9 shows the results.

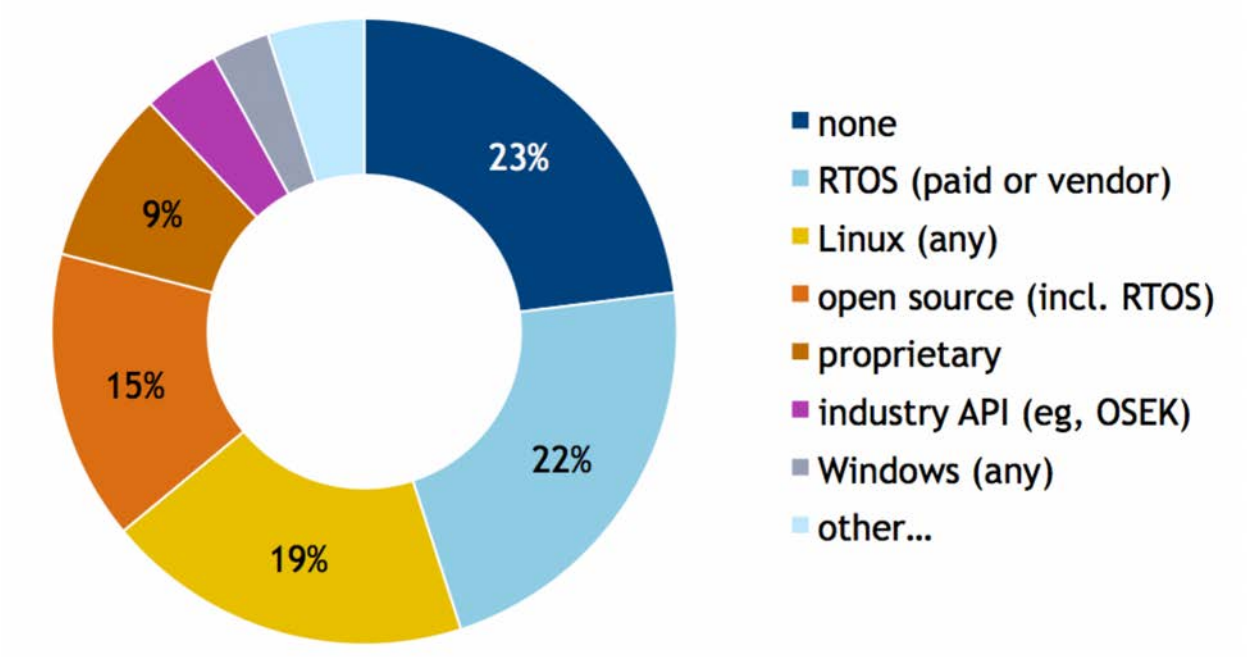


Figure 9. Type of Operating System on Primary Processor

Interestingly—even on the primary processor—the most popular type of operating system was "no operating system." The next largest group shown, "RTOS", aggregates those paying for a commercial real-time operating systems (e.g., VxWorks) with those using vendor-supplied RTOSes (typically from a processor maker, like TI). Following RTOSes was Linux, which was the third most popular type of operating. And after that are the open source operating systems (e.g., FreeRTOS) that don't have any licensing fees. Adding "proprietary" (i.e., company-internal) operating systems to the above brings the percentage of all designs covered to nearly 90%.

We can get some sense of the range in the architecture of embedded systems by comparing the rankings of the five most popular operating system choices based on the number of processors. As shown in Table 1, the percentage of designers writing their own "proprietary" operating system is about the same (9-10%) regardless of processor count. But Linux clearly becomes a much more popular choice (up from 13% to 32%) as the number of processors increases; while "open source" and "no operating system" become less popular.

1 processor	2-3 processors	4+ processors
none (33%)	RTOS (24%)	Linux (32%)
RTOS (18%)	none (22%)	RTOS (25%)
open source (18%)	Linux (17%)	none (11%)
Linux (13%)	open source (16%)	open source (9%)
proprietary (10%)	proprietary (9%)	proprietary (9%)

Table 1. Primary Operating System Choices by Number of Processors

Internet/Connectivity

As shown in Figure 10, the percentage of embedded systems designs that were expected to be always or sometimes connected (directly or indirectly) to the Internet was remarkably high, at nearly 60%.[14] The means of connection to other computers was indicated to be predominantly via wired network interfaces. However, about half of current designs had one or more wireless network interfaces.

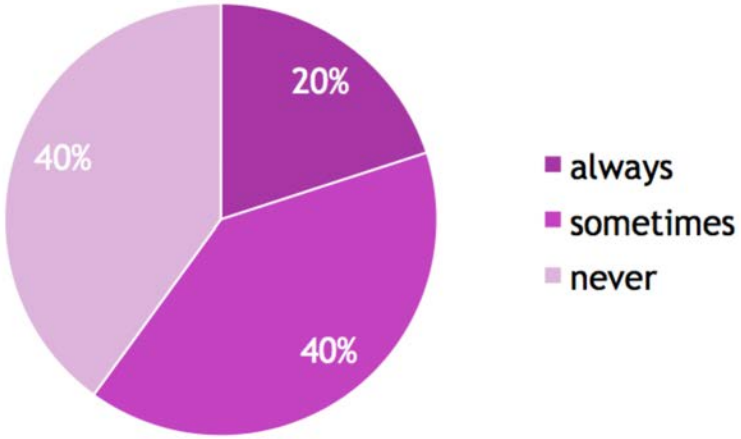


Figure 10. Frequency of Internet Connections

Of these, 38% said users of the product would be able to interact with it via a Web browser and 39% (not always the same respondents) said there would be an associated App (such as for iOS or Android).

[14] The percentage here reflects the weighted re-treatment of a small number (4% or 77) of "I don't know." responses into the three other categories. In so doing, we presume the design of these particular systems was in an earlier stage than others but that about the same percentages of "Always," "Sometimes," or "Never" responses would eventually apply. Without this, the "Always" + "Sometimes" total is 57% rather than 60% (and the total in the associated figure would be 96%). However, the set of "Internet-connected" designs analyzed elsewhere in this report omit these 77 respondents. Each of these choices was believed to offer the best-fit reflection of the industry.

Programming Languages

It's typically the case that multiple programming languages are used on a single project. For example, nearly every design will require at least one engineer to write at least a little bit of assembly language code. The trend toward multiple processors likely exacerbates the use of multiple languages as, for example, the primary processor might run Linux with applications written in Java or C++ while a sea of microcontrollers supporting it might each be coded entirely in C.

We sought some clarity by asking a straightforward question about the one "primary" programming language on their current project. As shown in Figure 11, nearly 95% of embedded programmers wrote the majority of their code in C or C++. Every other programming language was in the noise, at less than 1% each.

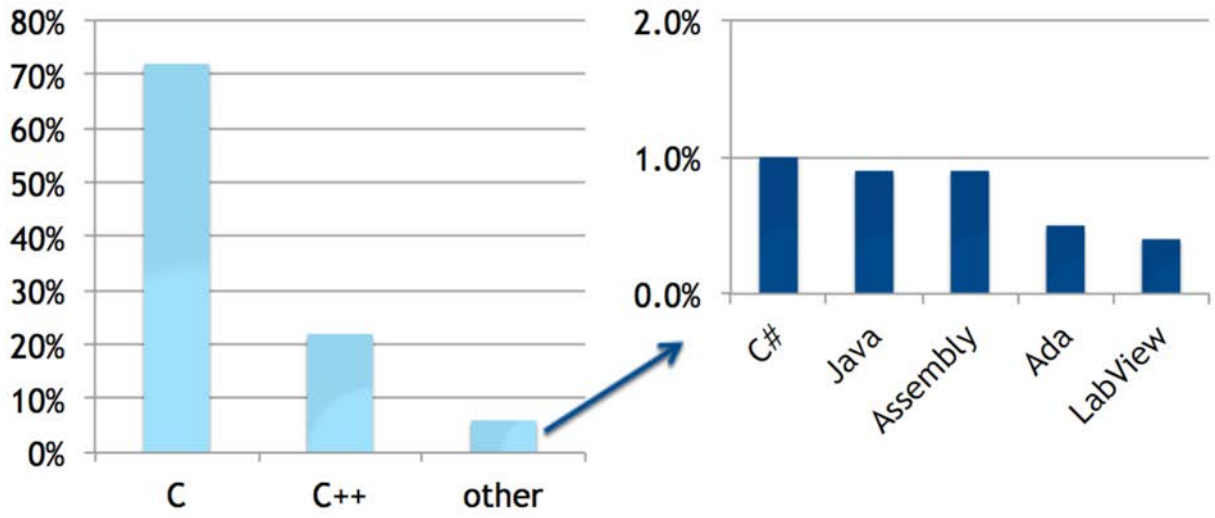


Figure 11. Primary Programming Language in Embedded Systems Designs

Note that C++ stole some share from C as the number of processors increased, with 34% of respondents designing a system with 4 or more processors indicating that C++ was the primary language—vs. just 15% in single-processor designs.

Software Development Processes

Figure 12 and Figure 13 show the percentage use of a set of well-known software development process steps within the embedded systems design community.

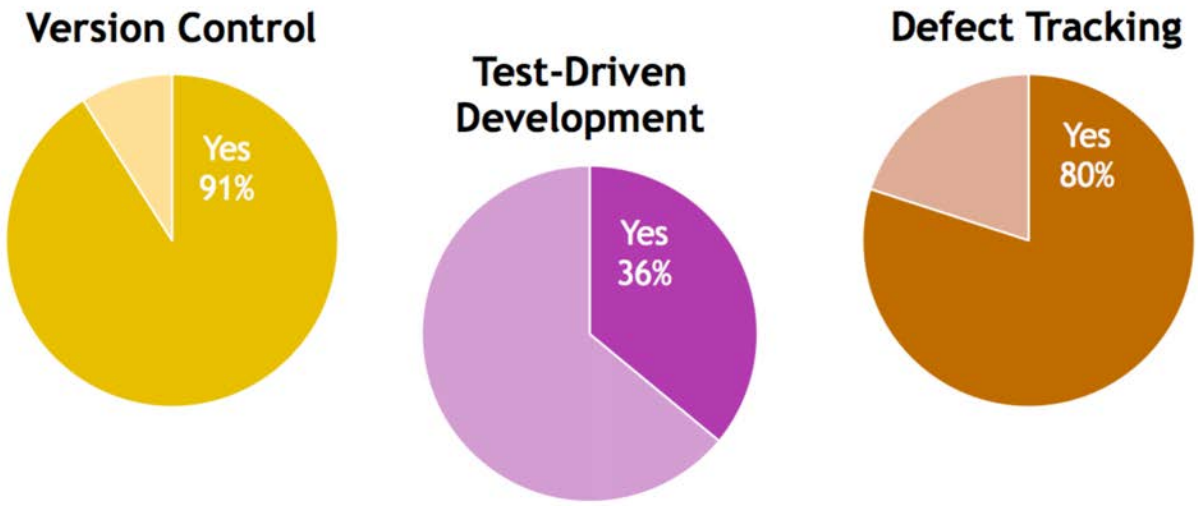


Figure 12. Percentage Use of Version Control, TDD, and Defect Tracking

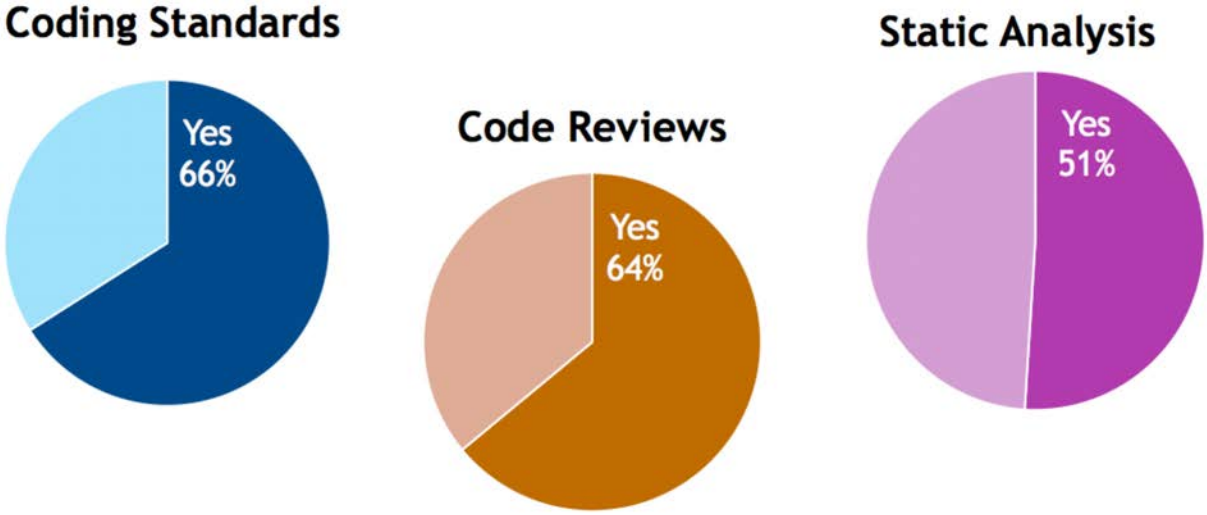


Figure 13. Percentage Use of Coding Standards, Code Reviews, and Static Analysis

Safety Analysis

The first issue to note about safety is the relatively high percentage of embedded systems that could—in the worst case—kill or injure one or more people. This and related data is broken down in Figure 14.

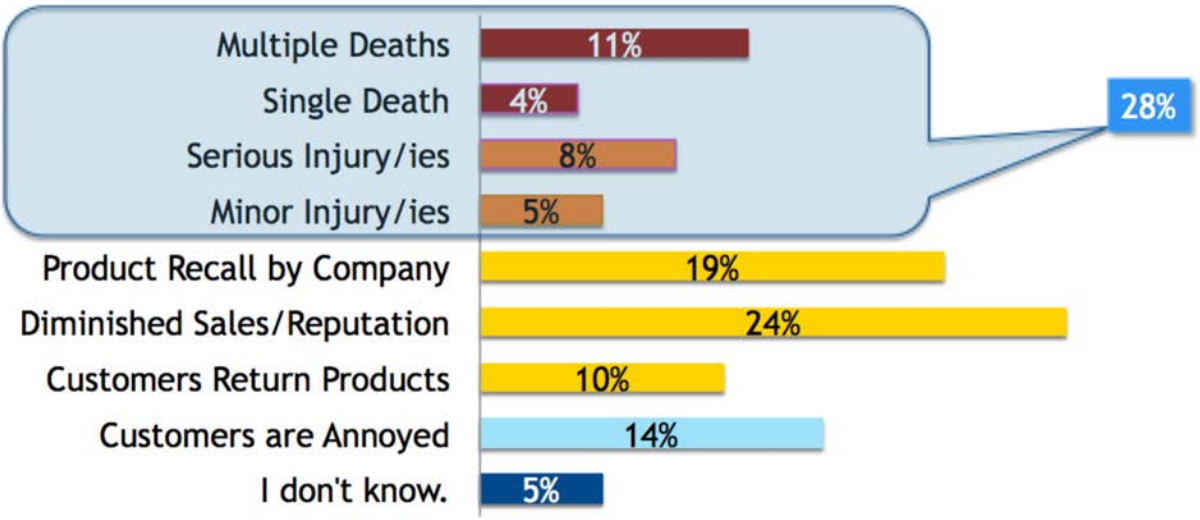


Figure 14. Worst-Case Possible Outcome in the Event of a Malfunction

We now analyze the survey data from the subset of professional embedded systems designers who could potentially end up with blood on their hands. More than 70% of these potentially dangerous products are targeted at one of four industries: medical devices, industrial controls, automotive systems, and defense/aerospace.

Safety-Related Practices

Coding Standards

Some good news is that compared with all embedded systems designers, the subgroup that is designing potentially dangerous products was more likely to have a written coding standard that applies to their product. Unfortunately, however, 17% of dangerous system designers do not have a written coding standard. And there are also risks in the way coding standards are being adopted and enforced.

Figure 15 shows less than half of the coding standards that are in place for dangerous products are based on standards specifically promoting safer systems: beginning from the MISRA, Barr Group, High Integrity, or JSF standards for C/C++. In our experience as consultants we have found that proprietary coding standards are almost always ad-hoc and highly unlikely to promote safer programming practices.

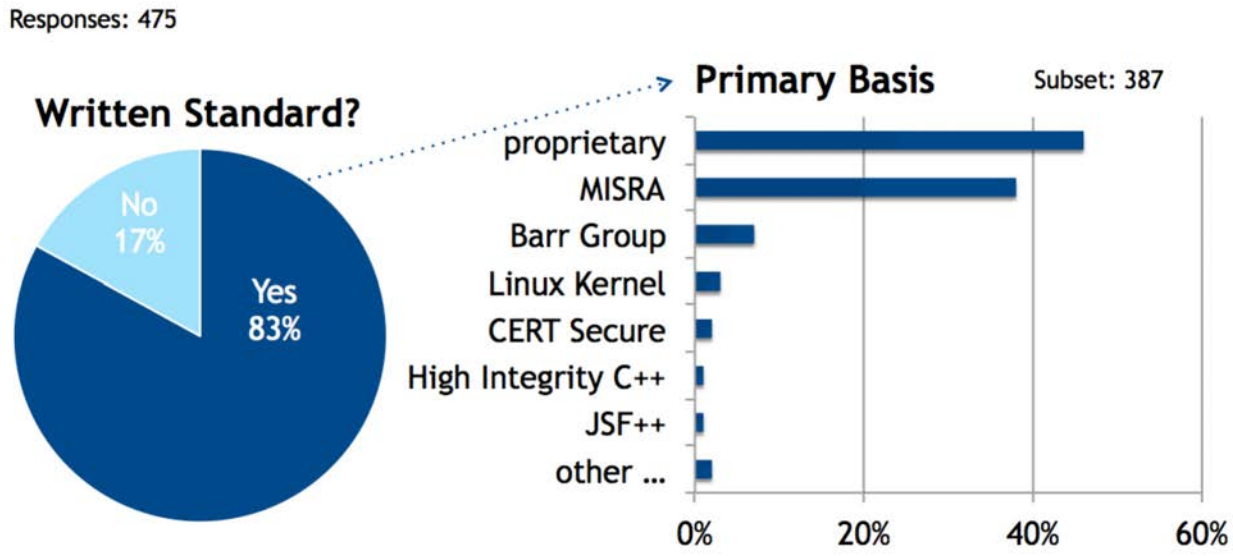


Figure 15. Primary Bases for Coding Standards Used in Safety-Critical Products

Furthermore, as shown in Figure 16, enforcement of coding standards is too lax.

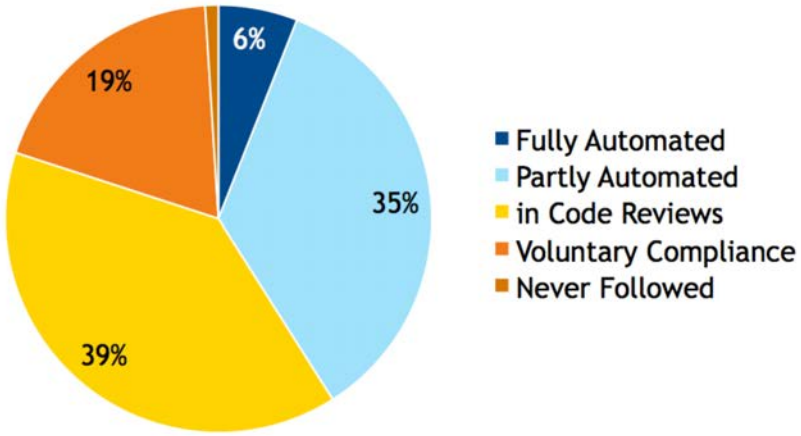


Figure 16. Enforcement of Coding Standards in Safety-Critical Products

Static Analysis

Static analysis tools are software programs that automate the process of scanning source code for potential bugs as well as violations of best practices. One of the most widely used of these tools, called PC/Lint, costs just a few hundred dollars to purchase. In addition to alerting programmers to potential problems in a repeatable and impartial way, static analysis tools can also be used to automate enforcement of many of the rules in coding standards.

Overall, slightly more than half of survey respondents indicated that their project's source code is run through one or more static analysis tools. Some good news was that the subgroup that is designing potentially dangerous products was more likely to use static analysis. Unfortunately, 32% of all dangerous system designers were not using static analysis. That's obviously a huge issue: people could be killed or injured by bugs that could have been easily and inexpensively flagged by static analysis.

Figure 17 breaks down the non-use of static analysis tools according to worst-case risk. Ideally this curve would not only trend downward to the right (as it does), but also reach 0% at or before the "one death" column. That around a quarter of the embedded systems that could kill are being programmed without static analysis as a step in the software development process is scary.

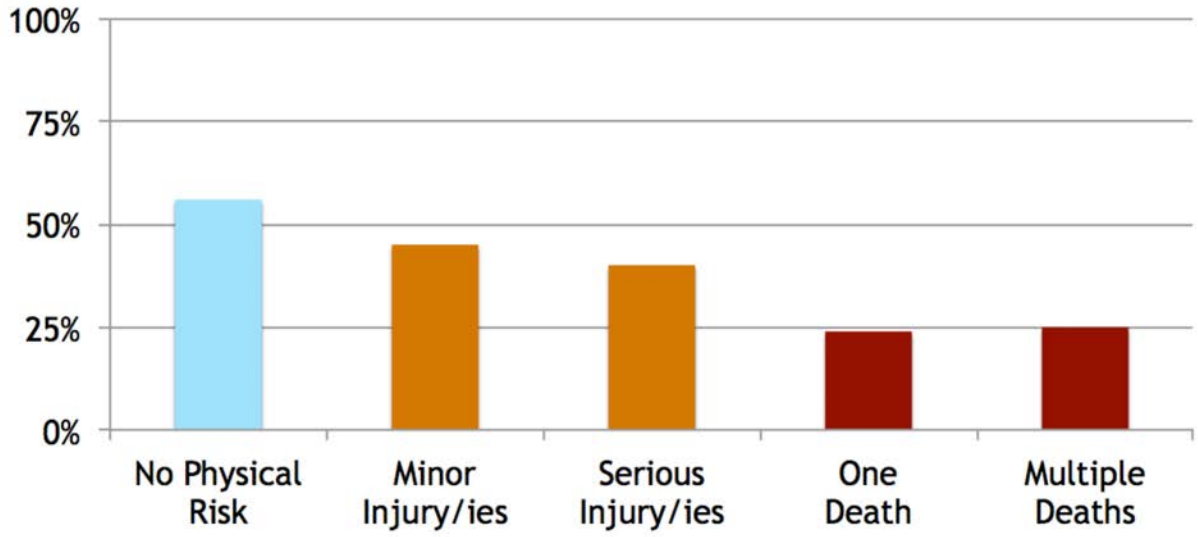


Figure 17. Percentage Non-Use of Static Analysis by Safety Risk Category

Code Reviews

Overall, nearly 60% of the designers of potentially dangerous systems said peer code reviews were either a regular process step on their current project or that pair programming was used. That's great news, as code reviews are well known to be one of the most cost-effective techniques for finding and fixing bugs in software.

Unfortunately, though, 25% of the developers of dangerous products said they rarely, if ever, perform code reviews in any way! And another 16% said they only do code reviews for some modules or when problems arise. Here again people could be killed or injured by bugs that might have been easily spotted in a code review.

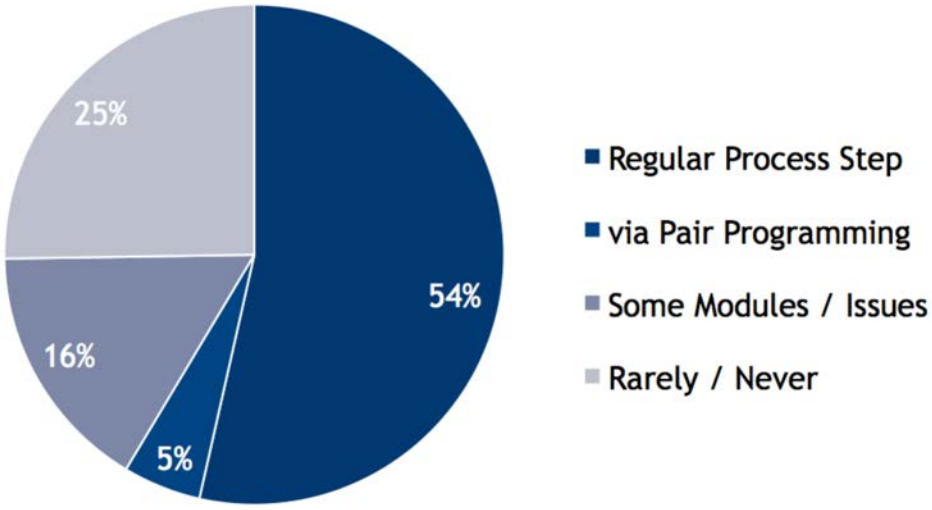


Figure 18. Percentage Use of Peer Code Reviews in Safety-Critical Products

Defect Tracking

Incredibly, 12% of respondents designing products that could kill or injure one or more people did not have any formal process or system in place to track known defects in their design. This is irresponsible behavior. No reasonably complex system can be completely free of bugs and defect-tracking need not be more difficult to setup than a spreadsheet or small database.

Safety Standards

There are a variety of industry-specific and general safety standards, such as MISRA's *Development Guidelines for Vehicle Based Software* and ISO-26262 for the automotive industry or IEC 61508. Broadly speaking, such safety standards and guidelines describe relevant best practices for designing safer systems as well as procedures for documenting when and how the practices were performed.[15]

In some cases, such as with the U.S. FDA's 510(k) guidelines for medical devices, products containing electronics and software cannot be legally sold in a country or region unless the developers conformed to the norms of the relevant safety standard.

One of the surprises in this year's survey data was that a large number of the designers of safety-critical systems are not following any relevant safety standard. Figure 19 has the numbers.

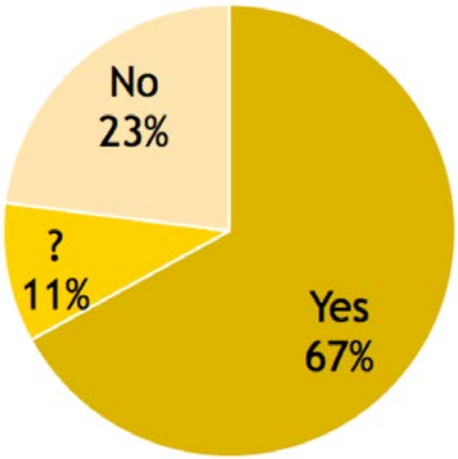


Figure 19. Percentage Use of Safety Standards in Safety-Critical Products

[15] The principle is similar to ISO-9001 in manufacturing: repeatable processes, properly executed, drive up consistency of outcomes. However, the design electronics and software is quite different from the manufacture of widgets in a factory. Thus the details

Testing

Testing of embedded systems takes many forms, from unit testing of individual software modules, to hardware-in-the-loop simulation that combines the full software on a hardware test-bed, and finally system-level testing of all of the components and their interactions. Of the available testing techniques, two are worthy of special mention in the context of safety-critical systems: test-driven development and regression testing.

Test-driven development (TDD) is a powerful technique for building reliable software that involves developing the test harness for each software module in parallel with writing the code that implements the actual functionality. A major benefit of TDD is that the library of test code grows as the product comes together and this testing code can be utilized to ensure that later changes in the functional code don't inadvertently break the system. As shown in Figure 20, only around a third of the designers of potentially dangerous embedded systems were employing TDD.

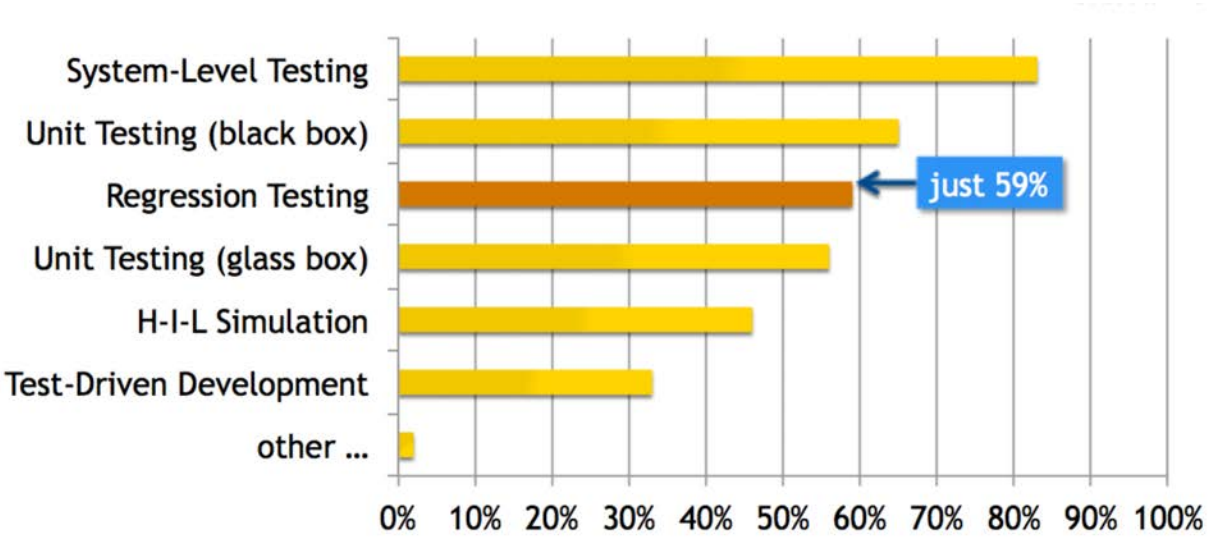


Figure 20. Types of Testing Performed on Safety-Critical Products

Regression testing is a powerful technique that generally ensures that the quality and reliability of a product can only increase over time (i.e., does not regress). In a nutshell, the method is to develop a large library of tests and to then retest each software upgrade by running all of the tests in the library.[16] Furthermore, each time a bug is found and killed it is a best practice to add new regression tests to the library to detect that type of bug should it recur in any future version.

There is no other type of testing that can ensure a monotonic increase in quality. Therefore, regression testing is an important software development step for ensuring safety. Thus it is concerning that only about 59% of the designers of embedded systems that could kill or injure people were using regression testing.

Findings

<u>There is a large opportunity to easily improve the safety of embedded systems by more broadly using well-known software development best practices.</u>

It is never sufficient to declare a system safe simply because certain processes and/or tools are used in the software development. There is much more to safety than process, including the architecture of the system. A team should also develop a formal written safety case to document the various design aspects that ensure that neither death nor injury can occur.

[16] This technique can be utilized in conjunction with the unit tests produced in the TDD process, though it does not require the use of TDD.

That said, certain best practice software development processes and tools are widely recommended and/or mandated by safety standards, including F.D.A. 510(k) guidelines for medical devices sold in the United States and the MISRA and ISO-26262 safety standards in the automotive industry.[17] And this is for a good reason: the use of processes such as coding standards, static analysis, and code reviews are—for example—well-studied, cost-effective techniques that can prevent and/or detect bugs before they are able to endanger product users.

This year's survey provides ample evidence that too many designers of safety-critical embedded systems are either not using some of these best practice process steps at all or aren't properly implementing those steps. Specifically, use of version control should be universal for all embedded systems designers. Likewise, keeping a database of known defects should be universal. The same is also true of use of static analysis tools and code reviews. And yet the failings of safety-critical system designers clearly go well beyond those to also include: not universally adopting or enforcing bug-killing coding standards; not universally setting up and using a testing system with quality-enhancing properties a la TDD and/or regression testing.

In some cases the failure of embedded systems designers to take these reasonable steps during development software may be indicative of engineering malpractice. As long as the current state of affairs persists, there will be many people utilizing and/or in the vicinity of unsafe devices and some of these people could be injured or killed by easily preventable or detectable bugs.

---

[17] Many embedded software engineers are familiar with the MISRA-C coding guidelines. However, the same automotive safety organization earlier published a set of *Development Guidelines for Vehicle Based Software* of which the better known C and C++ coding guidelines are a mere accompanying detail.

Safety practices are not clearly better in the automotive industry than in the medical device industry even though many more lives are at risk with automotive failures.

Data collected by this survey provides evidence that products designed for the medical device industry are not always designed according to software development best practices. Fortunately, the failure of a medical device (e.g., a pacemaker or an insulin pump) is considerably more likely to injure or kill just one person (i.e., the attached patient) than the multiple people that could be in harms way in industries such as automotive systems, transportation equipment, and industrial controls. Indeed, "multiple deaths" was the worst-possible outcome selected by just 5% of medical device designers.[18]

By contrast, 33% of the designers of automotive systems indicated that multiple deaths could occur as a result of a malfunction. (Another 1% selected "single death".) Given that the worst-case is so much worse for a single automotive failure and that a larger percentage of automotive system designs could kill, one would hope that designers of automotive systems exercise even greater care than those in the medical industry. In fact, the data shows that the practices of automotive system designers is sloppier in some respects.

Table 2 compares the software development practices of medical device designers whose systems could kill with those of automotive system designers who could kill. Some practices, such as adoption and enforcement of bug-killing coding standards, the automotive device designers exercise greater care. Yet, with others, such as code reviews, the automotive device designers exercise less care.

[18] An additional 21% of medical devices could kill one person. And it is possible that the 5% who answered "multiple deaths" were thinking of a series of single-person deaths.

	Medical Devices	Automotive Systems
Is the product designed in accordance with a **safety standard**?	89%	88%
Does the software development team use a **version control** system?	100%	98%
Are peer **code reviews** always a part of the software development process?	81%	69%
Is the source code run through one or more **static analysis** tools?	80%	92%
Is **regression testing** among the types of testing performed?	68%	74%
Are known defects formally tracked (e.g., in a **bug database**)?	100%	95%
Is there a **written coding standard** that applies to the project?	87%	95%[19]
Is the applicable coding standard **enforced** in some way?	89%	94%

Table 2. Comparison of Process Steps Used by Designers of Potentially Deadly Products

Overall, the comparison seems to show that automotive system designers are ahead on some practices and behind on others. The problem, however, is that designers of systems that could kill multiple people really ought to be at or near 100% on all of these practices. This point is particularly timely as the automotive industry actively experiments with and increasingly deploys self-driving vehicles and related technologies.

[19] The overwhelming majority (83%) of them had based their standard on the MISRA-C guidelines.

Safety, like justice, must be seen to be present. The graver the risk, the greater the need for documentation (e.g., written safety case) and careful process. Many safety standards, such as ISO-61508, have the designers select the necessary "System Integrity Level" (SIL) based on the worst-case possible outcome and this then dictates the processes that must be employed by the design team. For example, according to he MISRA software development guidelines the designers of all SIL2 systems must perform code reviews and the designers of SIL3 systems must also perform static analysis.

It is unfortunate but a fact that the reliability and safety of software cannot be "tested in". Safety is not a feature that can be later bolted on to a product. Rather, the reliability of a system must be "baked in" from the beginning as it derives from the system and software architecture as well as the software development and testing processes that are employed to prevent and detect bugs as early as possible in the design process.

Security Analysis

About 60% of respondents said that security was a design requirement on their current project. Of these, the majority indicated that their current design needed to be more secure than their prior design.

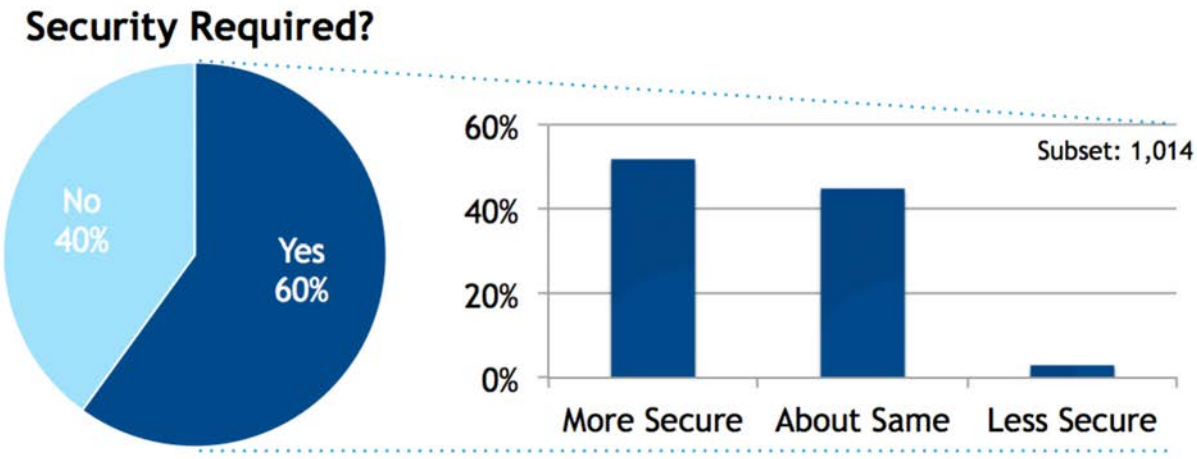


Figure 21. Percentage of Projects Having Security Requirements

Security-Related Practices

Primary Security Concerns

The 1,014 survey respondents who indicated there were design requirements relating to security on their project were asked to identify the one or more security concerns underlying these requirements. That is, what could go wrong if their device were successfully hacked. The results are shown in Figure 22.

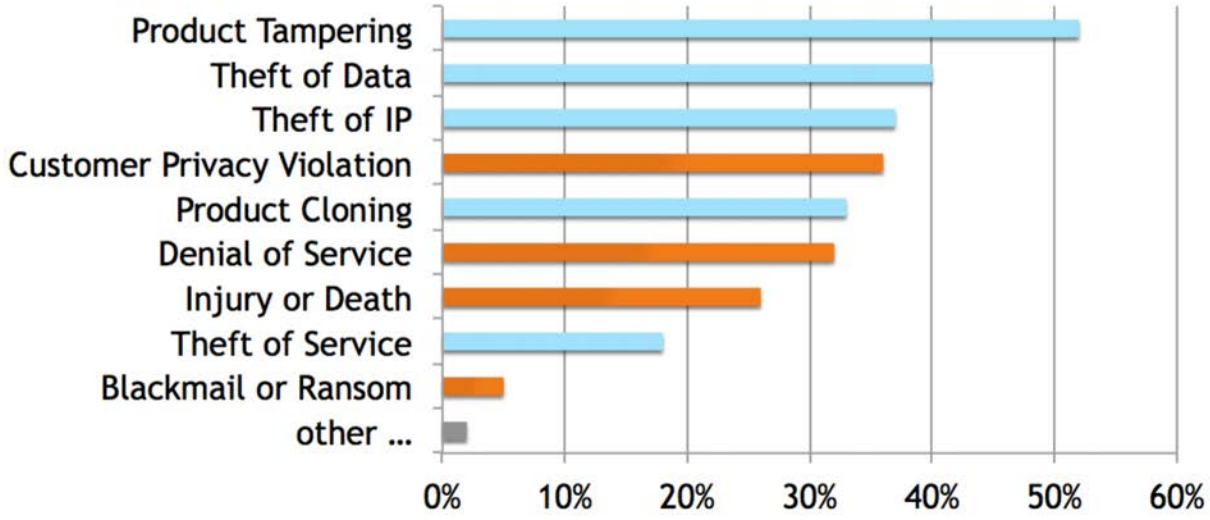


Figure 22. Ranking of Hacking Concerns of Products with Security Requirements

One interesting insight is that the highest-ranked security concerns were more likely to relate to the company that designed the product than to the users of the product. For example, a hacker who tampers with a product, steals the data or intellectual property of the company, or clones the product might be working for a competitor or otherwise able to undermine the business of the device maker. Violations of customer privacy, denials of service, injury/death, and blackmail/ransom were lower-ranking concerns for the designers.

Security Layers

The survey also asked those respondents with security concerns to select all of the security-related technologies they were using to improve the security of their products. For example, encryption of communications between the device and other system with which it will communicate. The results are shown in Figure 23.

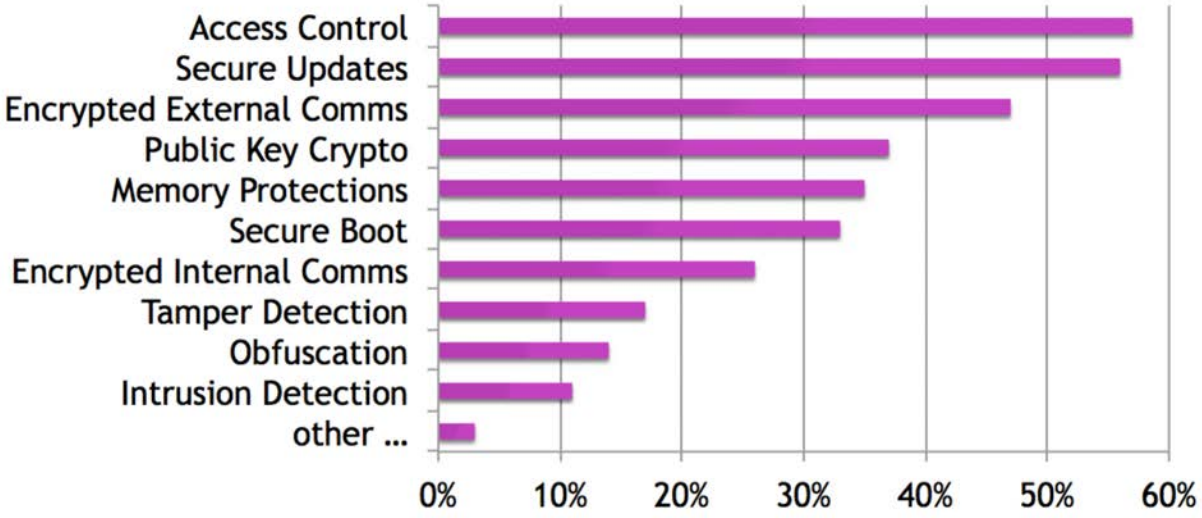


Figure 23. Percentage Use of Security-Related Technologies

We note that fewer than half of these systems would be encrypting external communications. And also that, although product tampering was the most common security concern for designers (see above), only about 17% of designs would have tamper detection technologies incorporated.

Security-Related Processes

Finally, we asked designers of systems with security as a design requirement what process steps they were taking to better secure their products. This could include the aforementioned software development best practices, such as code reviews and static analysis, in addition to threat and vulnerability analysis techniques and active testing techniques, such as fuzzing and penetration testing. The responses are shown in Figure 24.

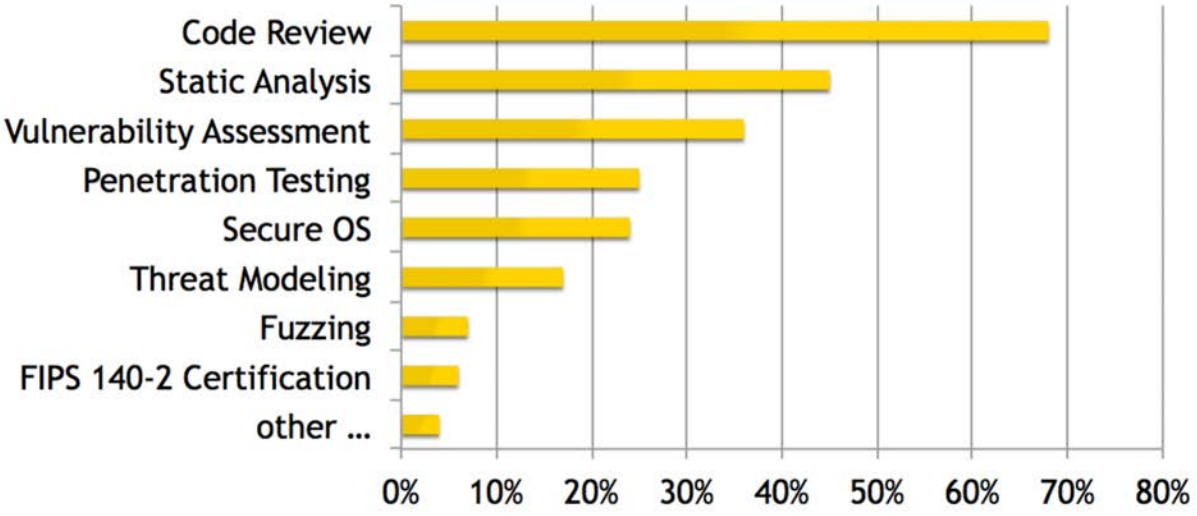


Figure 24. Percentage Use of Security-Related Processes

The Internet of Dangerous Things

Given that the majority of new embedded systems designs had connections to the Internet, it should not be surprising that a large number of safety-critical systems were going online too. Indeed, we identified a sizable subset (226) of respondents who were designing products that were both potentially injurious and on the Internet.[20] The percentages and numbers are broken down in Figure 25 and lead to a group we refer to henceforth as the Internet of Dangerous Things ("IoDT").

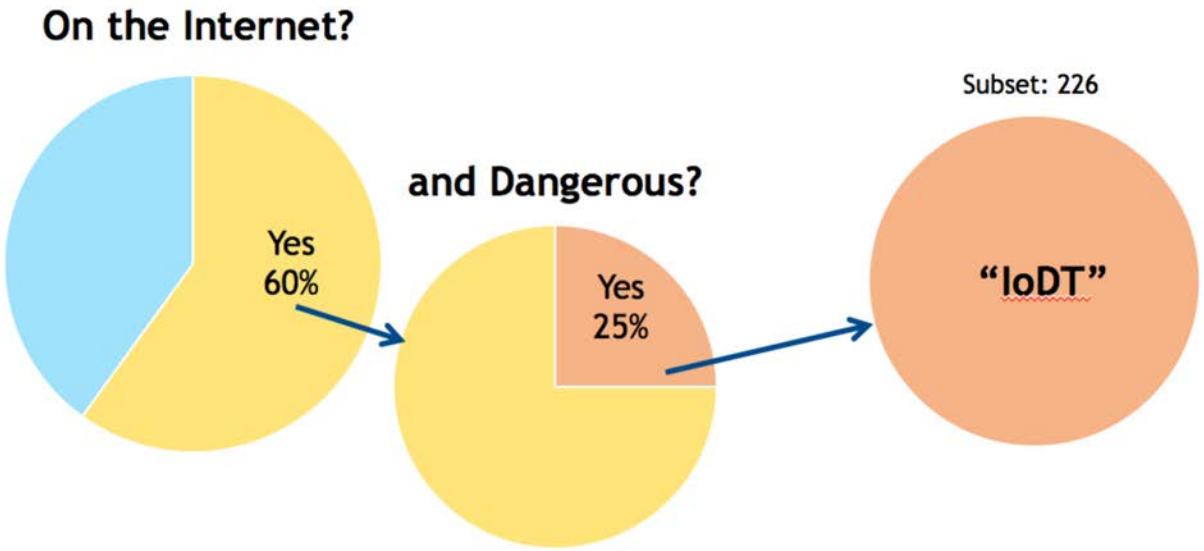


Figure 25. Percentage of Potentially Dangerous Systems with Internet Connections

Overall, about equal numbers could kill vs. merely cause injuries. A handful of industries were associated with more than two-thirds of the risk: medical devices (22%), industrial controls (21%), automotive systems (11%), consumer electronics (9%), and defense/aerospace (5%).

[20] Just 23% of these systems would be "always" on the Internet. Of course, systems that are on the Internet just some of the time can also be remotely hacked.

Findings

Broader use of software development best practices is also an opportunity to better secure the vast numbers of Internet-connected devices to come.

The security of a product depends in part on its reliability. For example, a medical device that can be made to malfunction by rapid pressing of keys could be attacked via that interface. Thus it is a security problem for the embedded systems industry as a whole that the best practice software development processes described above in the context of safety are not more widely utilized.

Figure 26 shows the rates of non-use of three best practices by the designers of potentially dangerous products that will be Internet-connected. Unbelievably, 37% of the designers of these systems either didn't have a written coding standard or did but didn't have any enforcement mechanism in place. With respect to peer code reviews, 24% never or rarely did them at all and another 18% said they did code reviews only sporadically and on some modules. Finally, more than a third didn't perform static analysis on their source code.

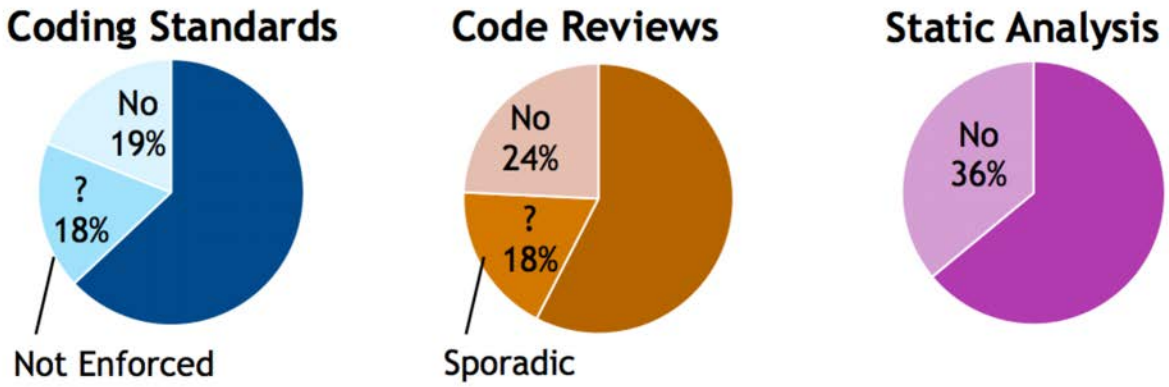


Figure 26. Percentage Non-Use of Best Practices on the Internet of Dangerous Things

Though there is obviously much more to designing a secure system than just following best practices for software development, these process steps represent a sort of low-hanging fruit for the industry to potentially inexpensively raise security for the Internet of Dangerous Things.

Designers of a remarkably large number of potentially dangerous embedded systems are ignoring security even as they connect their products to the Internet.

Disturbingly, 22% of the designers of safety-critical systems that would be connected to the Internet said that security was not a "design requirement" at all on their project. This is clearly a serious issue. By the time of this survey, the Internet was well-known to be a dangerous place for computers—with examples including both military-grade attacks (e.g., Stuxnet) as well as newsworthy wired and wireless attacks on automobiles and medical devices.

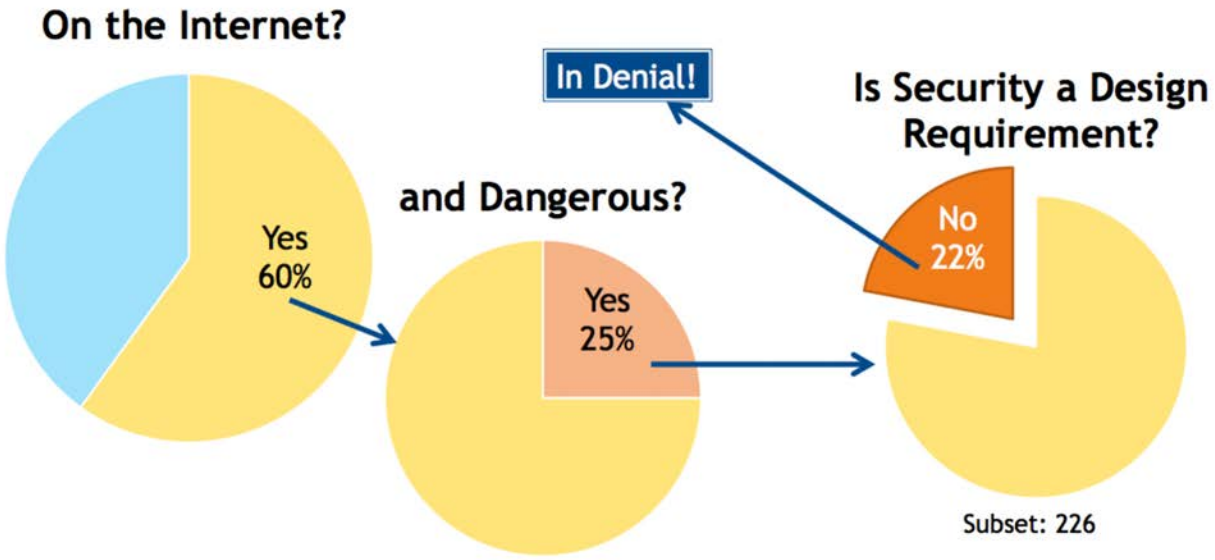


Figure 27. Percentage of Internet of Dangerous Things Designers Ignoring Security

Such a high level of ignorance and/or denial is alarming. What horrifying deadly disaster need occur before designers of Internet-connected products will begin to take security seriously?

Because the range of architectures and applications is large, there will never be a one-size-fits-all solution to the problem of securing embedded systems.

Unlike software designed for general-purpose computers, embedded software cannot usually be run on other embedded systems without significant modification. This is primarily because of the incredible variety in the underlying hardware. The hardware in each embedded system is tailored specifically to the application, in order to keep system costs low. As a result, unnecessary circuitry is eliminated and hardware resources are shared wherever possible.

By definition all embedded systems contain at least one processor and software, but increasingly the number of microcontrollers and/or processor cores is itself a point of architectural differentiation. Only about a third of systems have just one processor while nearly a quarter have 4 or more. And each of these processors can be chosen from across dozens of popular semiconductor makers and instruction set families.

The rest of the embedded hardware is equally unique. The inputs to an embedded system usually take the form of sensors and probes, communication signals, or control knobs and buttons. The outputs are typically displays, communications signals, or changes to the physical world. But these inputs and outputs vary incredibly widely across product types and target industries.

Architectural variation of the hardware and software is the result of many competing design criteria. Each embedded system is a product that must meet a completely different set of requirements, any or all of which may affect the compromises and tradeoffs made during the development of the product. For example, if the system must have a production cost below $10, then other things—like processing power, memory, and system reliability—may need to be sacrificed in order to meet that goal.

As illustrated by the data summarized in Figure 28, the hypothetical designer of a "one-size-fits" all security solution for embedded systems would be faced with a daunting challenge: the potential attack surfaces, number of processors to defend, and operating system platforms are among many factors that make such an easy solution impossible. The solution space for embedded security is thus vast and unlikely to ever constitute an efficient market.

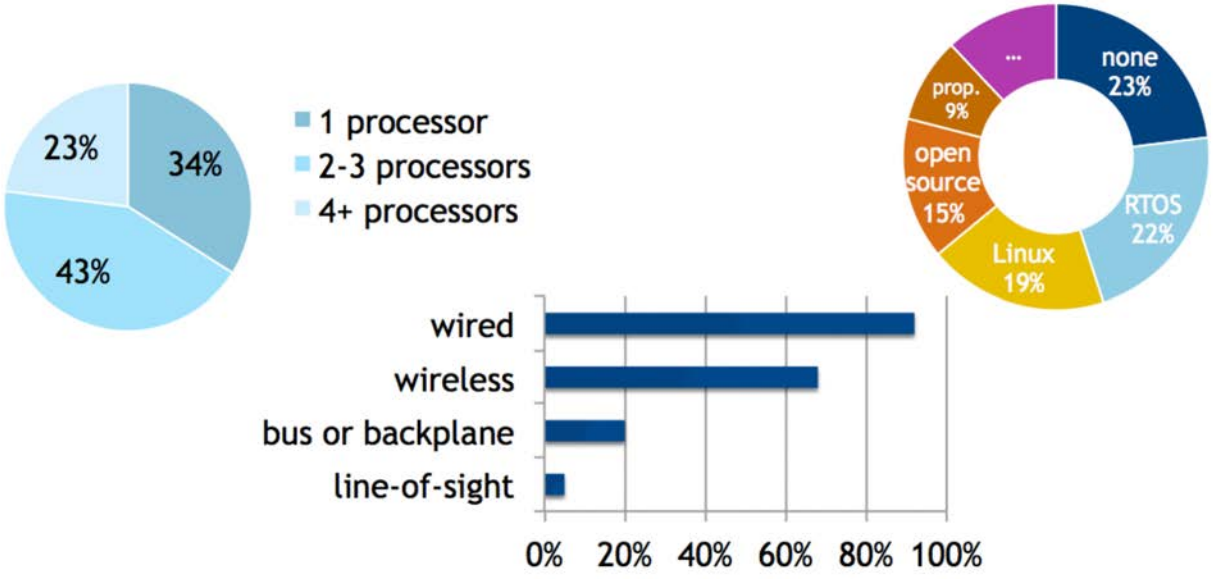


Figure 28. Diversity of Embedded Systems Hardware and Software Architectures

Barr Group

2017

Embedded Systems Safety & Security Survey

Appendix A: Survey Questions as Asked

2017 Embedded Systems Survey

Thanks for taking a few minutes to help with this important annual survey of worldwide industry trends.

* 1. How much professional experience (paid work, not counting academic work) do you have in the field of embedded systems design?

- I have no professional experience in embedded systems design.
- 1-9 years
- 10-19 years
- 20-29 years
- 30+ years

2017 Embedded Systems Survey

* 2. What is your primary professional role in the design of embedded systems?

- ○ Engineer with software/firmware design focus
- ○ Engineer with hardware/electronics design focus
- ○ Engineer who regularly does both software design and hardware design
- ○ Engineer with system-level or architecture-level focus
- ○ Manager with direct oversight of one or more design projects
- ○ Executive or manager with no direct oversight of design projects
- ○ I am primarily involved in ensuring product quality (e.g., testing and validation)
- ○ I work in academia and/or primarily teach
- ○ Other (please specify)

2017 Embedded Systems Survey

Answer all remaining questions about a single embedded systems design project you are currently involved with.

* 3. Which one of the following product categories best applies to your current project?

- Automation or Industrial Controls
- Consumer Electronics
- Gaming Devices or Systems
- Communications and Networking
- Internet of Things
- Scientific Instruments (e.g., oscilloscopes, colorimeters)
- Automotive Systems
- Transit/Transportation (e.g., rail, bus, boat)
- Farming or Construction Equipment
- Medical Devices or Instruments
- Aerospace or Defense
- Computers and Related Peripherals
- Oil or Gas Extraction or Refinement
- Electricity Generation or Distribution
- Public Utilities or Municipal Government
- Home or Business Security Systems
- Audio/Video/Image Capture/Processing/Playback
- Banking or Finance
- Semiconductors
- I don't know.
- Other (please specify)

* 4. What is the nature of your current project?

- Software-only upgrade/refinement for existing product
- Upgrade/refinement of hardware and software for existing product
- Complete redesign of existing product
- Cost reduction of hardware for existing product
- A brand new type of product, mostly from scratch
- A brand new type of product, mostly based on an earlier product
- I don't know.

* 5. How many total processors (including microcontrollers and cores) do you expect to be included?

- There are no processors.
- 1 processor
- 2-3 processors
- 4+ processors
- I don't know.

* 6. At peak effort, how many people will be involved in writing embedded software for your current project?

- None, because there's no embedded software on this project.
- 1 programmer
- 2-4 programmers
- 5-9 programmers
- 10-19 programmers
- 20+ programmers
- I don't know.

* 7. What type of primary operating system do you expect to run on the main processor?

- A commercial (i.e., one you pay for) RTOS (e.g., MicroC/OS, VxWorks)
- A chip-vendor RTOS (e.g., TI/BIOS, MQX)
- A free RTOS (e.g., eCOS, FreeRTOS)
- Android
- Another flavor of Linux (whether commercial or free)
- A flavor of Microsoft Windows (desktop or embedded)
- A state machine framework (e.g., Quantum Platform)
- An industry-standard API (e.g., AUTOSAR/OSEK)
- A proprietary operating system (i.e., an in-house design)
- No operating system
- I don't know.
- Other (please specify)

8. If you know, what types of external connections will your current project have? (select all that apply)

- [ ] One or more wired connections (e.g., Ethernet, USB, CAN, other serial/parallel interface)
- [ ] One or more wireless connections (e.g., 802.11, Bluetooth, radio, cellular, satellite)
- [ ] One or more line-of-sight connections (e.g., IrDA)
- [ ] One or more bus or backplane connections (e.g., PCI, SATA, VME)
- [ ] Other (please specify)

* 9. When, if at all, will your current project be connected (directly or indirectly) to the Internet?

- It will never be connected to the Internet
- It will sometimes be connected to the Internet
- It will always be connected to the Internet
- I don't know.

10. If you know, how will users interact with your current project? (select all that apply)

- [ ] Via a graphical user interface (a.k.a., GUI)
- [ ] Via physical controls (knobs/buttons/switches)
- [ ] Via an App (e.g., for iOS or Android)
- [ ] Via a web browser (i.e., there will be an embedded web server)
- [ ] Other (please specify)

2017 Embedded Systems Survey

Still thinking about the same current embedded systems design project you are personally involved with...

* 11. What is the primary programming language for your current project?

- C
- C++
- Assembly
- Java
- C# / .NET
- Ada
- LabView
- I don't know.
- Other (please specify)

* 12. Does your team use a version control system?

- Yes
- No
- I don't know.

* 13. Are peer source code reviews a part of the software development process?

- ( ) We have a process in place that ensures regular code reviews for all code.
- ( ) We perform continuous peer review by pair programming.
- ( ) We conduct code reviews for some modules and/or when there is a problem.
- ( ) In theory we are supposed to hold code reviews, but we hardly or never actually do them.
- ( ) Code reviews are not part of our process and/or there is only one programmer.
- ( ) I don't know.

* 14. Is source code run through one or more static analysis tools (e.g., PC/Lint or Coverity)?

- ( ) Yes
- ( ) No
- ( ) I don't know.

15. If you know, what kinds of testing will be performed? (select all that apply)

- [ ] Test-Driven Development
- [ ] White Box Unit Testing (includes testing of all internal states of the unit)
- [ ] Black Box Unit Testing (ignores internal states, focusing only on outputs)
- [ ] Regression Testing
- [ ] System Testing
- [ ] Hardware-in-the-Loop Testing
- [ ] Other (please specify)

* 16. Are known defects formally tracked (e.g., in a bug database or issue tracking system?

- ( ) Yes
- ( ) No
- ( ) I don't know.

* 17. Is there a written coding standard in place that applies to your current project?

- ( ) Yes
- ( ) No
- ( ) I don't know.

2017 Embedded Systems Survey

Still thinking about the same current embedded systems design project you are personally involved with...

* 18. What is the primary basis of the coding standard that applies to your current project?

- MISRA's Guidelines for Critical Systems for C or C++
- CERT's Secure Coding Standards for C, C++, or Java
- Lockheed's Joint Strike Fighter Standard for C++ (JSF++)
- High Integrity C++ Standard
- Barr Group's Embedded C Coding Standard
- Linux Kernel Coding Standard
- A proprietary coding standard (i.e., in-house developed)
- I don't know.
- Other (please specify)

* 19. How is the relevant coding standard enforced on your current project?

- Enforcement is automated and non-compliant code cannot be checked-in.
- Enforcement is partly automated with static analysis tool use.
- Enforcement is one of the issues checked during code reviews.
- There is no enforcement mechanism, though some programmers voluntarily comply.
- Our coding standard is a meaningless "write-once/read-never" document.
- I don't know.

2017 Embedded Systems Survey

Still thinking about the same current embedded systems design project you are personally involved with...

* 20. Is security one of the design considerations on your current project?

- ◯ Yes
- ◯ No
- ◯ I don't know.

21. How do the security needs of your current project compare to your other recent projects?

- ◯ The current project needs to be more secure than other recent projects.
- ◯ The security needs are about the same as on other recent projects.
- ◯ It's okay if the current project is less secure than other recent projects.
- ◯ I don't know.

2017 Embedded Systems Survey

Still thinking about the same current embedded systems design project you are personally involved with...

* 22. What is the relative importance of security vs. meeting the schedule on your current project?

- ○ Security is much more important than meeting the schedule.
- ○ Security is more important than meeting the schedule.
- ○ Security and meeting the schedule are about equally important.
- ○ Security is less important than meeting the schedule.
- ○ Security is much less important than meeting the schedule.
- ○ I don't know.

23. If you know, what are the primary security concerns with your current project? (select all that apply)

- ☐ Product Cloning
- ☐ Theft of Intellectual Property
- ☐ Customer Privacy Violations
- ☐ Theft of Data
- ☐ Product Tampering
- ☐ Theft of Service
- ☐ Denial of Service
- ☐ Injury or Death
- ☐ Blackmail or Ransom
- ☐ Other (please specify)

24. If you know, which of the following security layers are used on your current project? (select all that apply)

- [ ] Non-Volatile Memory Protections
- [ ] Mechanical Tamper Detection
- [ ] Network Intrusion Detection
- [ ] Access Control (e.g., user authentication)
- [ ] Encrypted and Authenticated External Communications (e.g., SSL/TLS)
- [ ] Encrypted Internal Communications
- [ ] Secure Boot Process
- [ ] Secure Firmware Updates
- [ ] Public Key Cryptography
- [ ] Obfuscation
- [ ] Other (please specify)

25. If you know, which of the following processes are used<u>to increase security</u> on your current project? (select all that apply)

- [ ] Threat Modeling
- [ ] Code Review
- [ ] Static Analysis
- [ ] FIPS 140-2 Certification
- [ ] Fuzzing
- [ ] Secure Operating System
- [ ] Vulnerability Assessment
- [ ] Penetration Testing
- [ ] Other (please specify)

2017 Embedded Systems Survey

Still thinking about the same current embedded systems design project you are personally involved with...

* 26. If the product resulting from your current project malfunctioned, what is the worst possible outcome?

- Death of Multiple People
- Death of One Person
- Serious Injury of One or More People
- Minor Injury to One or More People
- Product Recall by Company
- Diminished Sales and/or Brand Reputation
- Customers Return Products
- Customers are Annoyed
- I don't know.

* 27. How do the reliability needs of your current project compare to your other recent projects?

- The current project needs to be more reliable than other recent projects.
- The reliability needs are about the same as for other recent projects.
- It's okay if the current project is less reliable than other recent projects.
- I don't know.

* 28. Will your current project be designed in accordance with a safety standard (e.g., FDA, DO-178, IEC-61508)?

- ◯ Yes
- ◯ No
- ◯ I don't know.

Name(s) of applicable standards, if any:

* 29. What is the relative importance of reliability vs. security on your current project?

- ◯ Reliability is much more important than security.
- ◯ Reliability is more important than security.
- ◯ Reliability and security are about equally important.
- ◯ Reliability is less important than security.
- ◯ Reliability is much less important than security.
- ◯ I don't know.

* 30. What is the relative importance of reliability vs. meeting the schedule on your current project?

- ◯ Reliability is much more important than meeting the schedule.
- ◯ Reliability is more important than meeting the schedule.
- ◯ Reliability and meeting the schedule are about equally important.
- ◯ Reliability is less important than meeting the schedule.
- ◯ Reliability is much less important than meeting the schedule.
- ◯ I don't know.

2017 Embedded Systems Survey

* 31. Approximately how many total people work at your company (across all locations)?

- 1-9 people
- 10-99 people
- 100-999 people
- 1,000-9,999 people
- 10,000+ people
- I don't know.

* 32. Approximately how many engineers (of any type) work at the company?

- 1-9 engineers
- 10-99 engineers
- 100-999 engineers
- 1,000+ engineers
- I don't know.

33. Which of the following countries or regions best describes where you currently reside?

- United States
- Canada
- Mexico
- *Rest of North America*
- Central America
- Brazil
- Argentina
- *Rest of South America*
- United Kingdom
- Scandinavia
- Germany
- Italy
- Spain
- France
- Eastern Europe
- Russia
- *Rest of Europe*
- Israel
- Middle East
- Australia
- New Zealand
- India
- Korea
- China
- Singapore
- Taiwan
- Japan
- *Rest of Asia*
- Africa
- *Somewhere Else*

2017 Embedded Systems Survey

34. In which part of the United States do you currently reside?

Barr Group

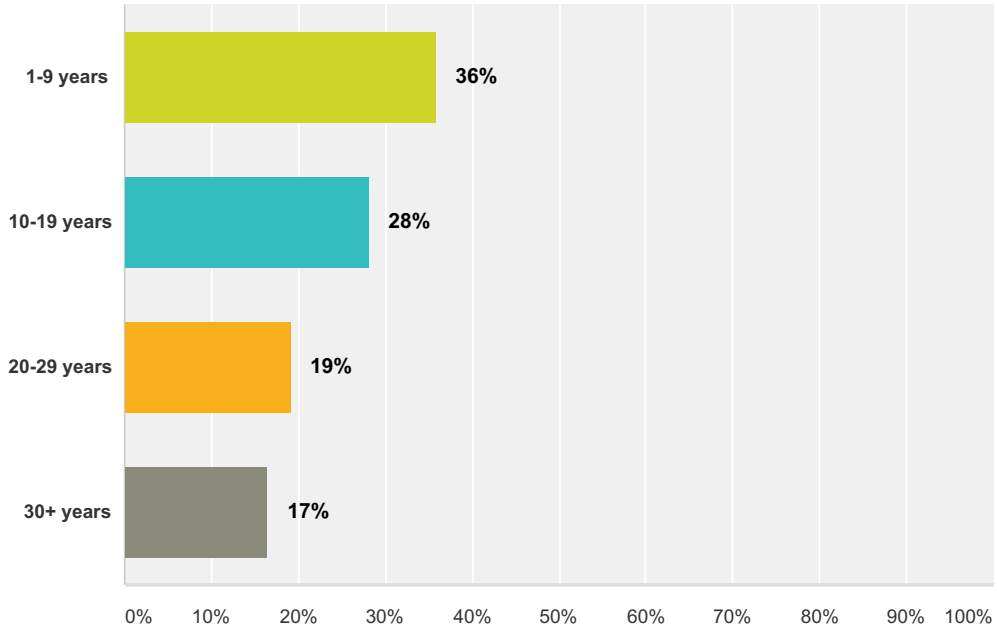
2017

Embedded Systems Safety & Security Survey

Appendix B: Qualified Responses as Received

Q1 How much professional experience (paid work, not counting academic work) do you have in the field of embedded systems design?

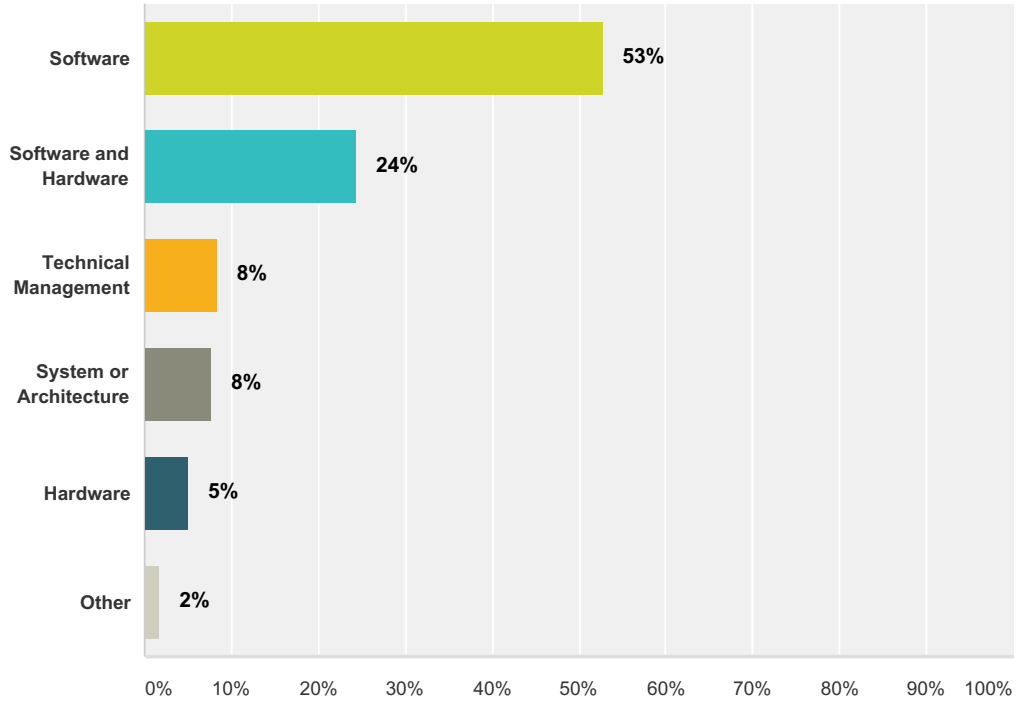
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
1-9 years	**36%**	621
10-19 years	**28%**	487
20-29 years	**19%**	332
30+ years	**17%**	286
Total		**1,726**

Q2 What is your primary professional role in the design of embedded systems?

Answered: 1,726 Skipped: 0



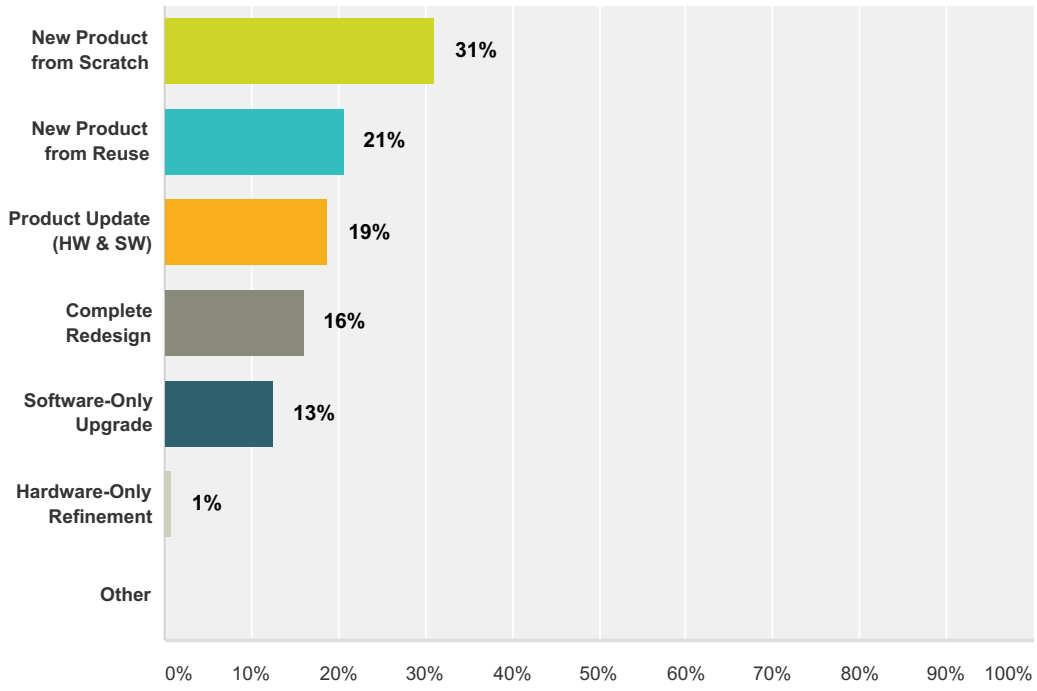
Answer Choices	Responses	
Software	53%	910
Software and Hardware	24%	420
Technical Management	8%	144
System or Architecture	8%	135
Hardware	5%	88
Other	2%	29
Total		1,726

#	Other (please specify)	Date
1	firmware engineer/architect/manager/director - done it all	1/27/2017 7:37 PM
2	Team Leader HW,SW&FW Design experience and team management	1/25/2017 5:49 PM
3	Engineer with hardware/electronics design focus AND low level firmware (works together)	1/25/2017 4:31 PM
4	Consultant /Functional Safety/ Process Development	1/19/2017 9:07 AM
5	Designed for 14 years+, now an Applications engineer	1/18/2017 9:47 PM
6	Was arch-level engineer, now Fed regulatory review of embedded systems	1/18/2017 7:03 PM
7	between contracts	1/18/2017 5:54 PM
8	Engineer turned Marketing	1/18/2017 5:27 PM

9	Product manager for control components	1/18/2017 5:11 PM
10	Manager, firmware, architecture, and electrical	1/17/2017 6:18 PM
11	Engineer who does system + hw + fw	1/16/2017 9:33 PM
12	FPGA Engineer by title, but I actually do a lot of FW and integration work	1/16/2017 8:04 PM
13	Continuous Improvement Lead focusing on development systems, processes and tools.	1/16/2017 7:54 PM
14	Mix of HW eng and system/project oversight	1/16/2017 7:27 PM
15	Works in Application Layer in Automotive Embedded System Features	1/15/2017 10:40 AM
16	Safety	1/14/2017 9:37 AM
17	Technology journalist	1/13/2017 6:43 PM
18	I m involved in evaluation of Supplier SW Embedded process (e.g. ASPICE, CMMI application)	1/13/2017 9:25 AM
19	Safety management; safety concept development; safety analysis	1/13/2017 6:25 AM
20	Engineer who regularly does both software design and hardware desin. Engineer with system-level or architecture-level focus	1/13/2017 3:26 AM
21	Currently technical sales	1/13/2017 2:23 AM
22	Engineer who regularly does software, firmware, hardware design and testing, system-level design, and managing projects (small team/organization)	1/12/2017 10:19 PM
23	design consulting	1/12/2017 10:02 PM
24	Firmware Developer without professional engineering designation.	1/12/2017 5:45 PM
25	Independent Consultant at a whole system level and capabilities in electronics and software.	1/12/2017 5:43 PM
26	Consultant	1/12/2017 4:54 PM
27	I wear a lot of hats, SW,FW,Arch, Systems, How To	1/12/2017 4:38 PM
28	Functional Safety	1/11/2017 9:41 PM
29	Architecture and Software/Hardware design	1/10/2017 7:00 PM

Q4 What is the nature of your current project?

Answered: 1,726 Skipped: 0

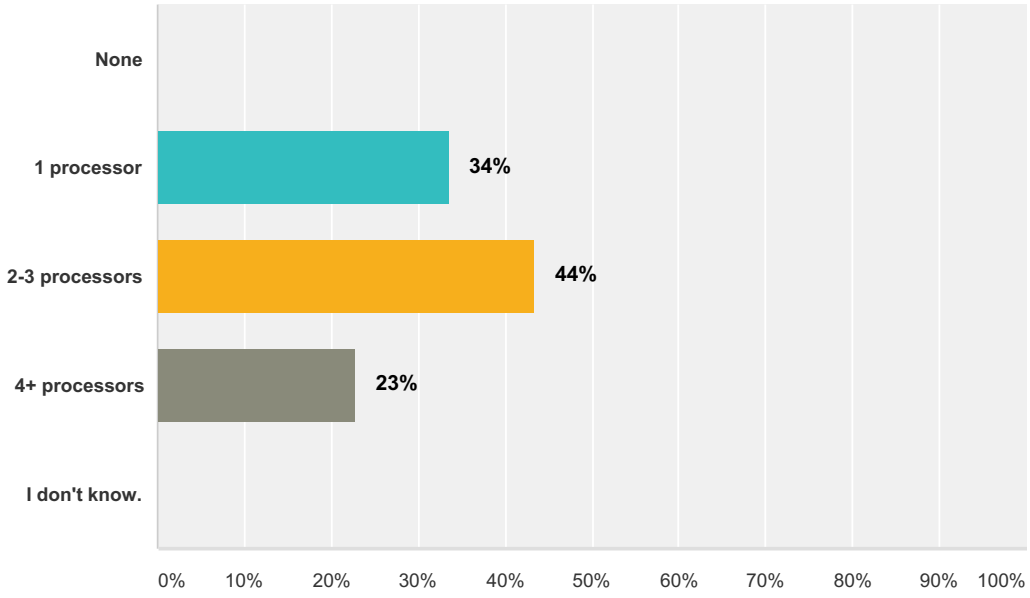


Answer Choices	Responses	
New Product from Scratch	31%	536
New Product from Reuse	21%	355
Product Update (HW & SW)	19%	326
Complete Redesign	16%	278
Software-Only Upgrade	13%	216
Hardware-Only Refinement	1%	15
Other	0%	0
Total		**1,726**

#	Other (please specify)	Date
	There are no responses.	

Q5 How many total processors (including microcontrollers and cores) do you expect to be included?

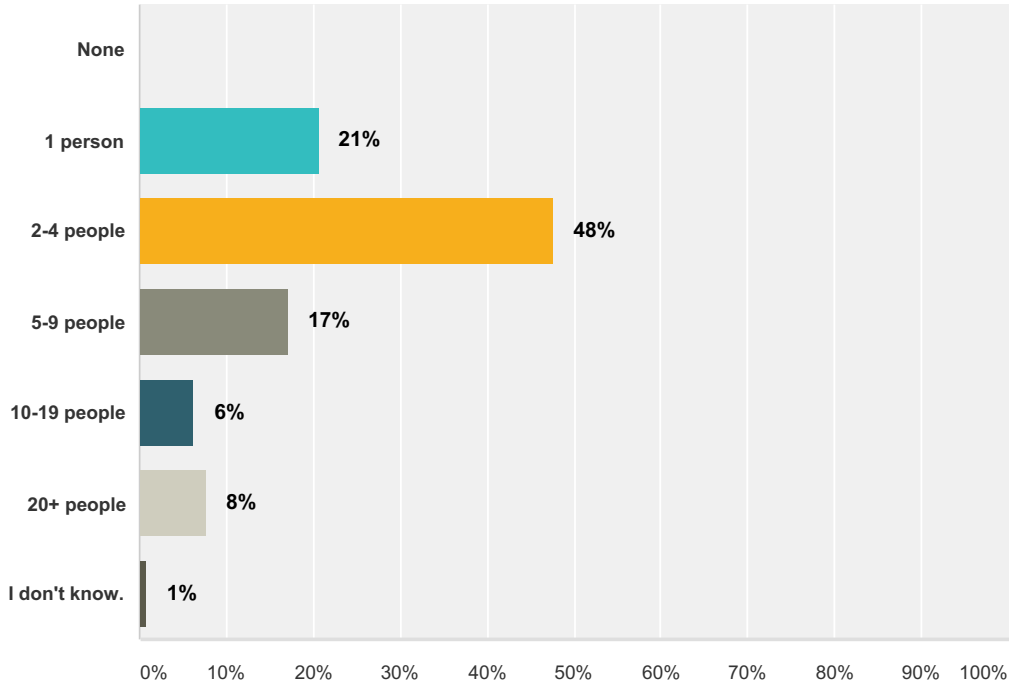
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
None	0%	0
1 processor	34%	581
2-3 processors	44%	751
4+ processors	23%	394
I don't know.	0%	0
Total		1,726

Q6 At peak effort, how many people will be involved in writing embedded software for your current project?

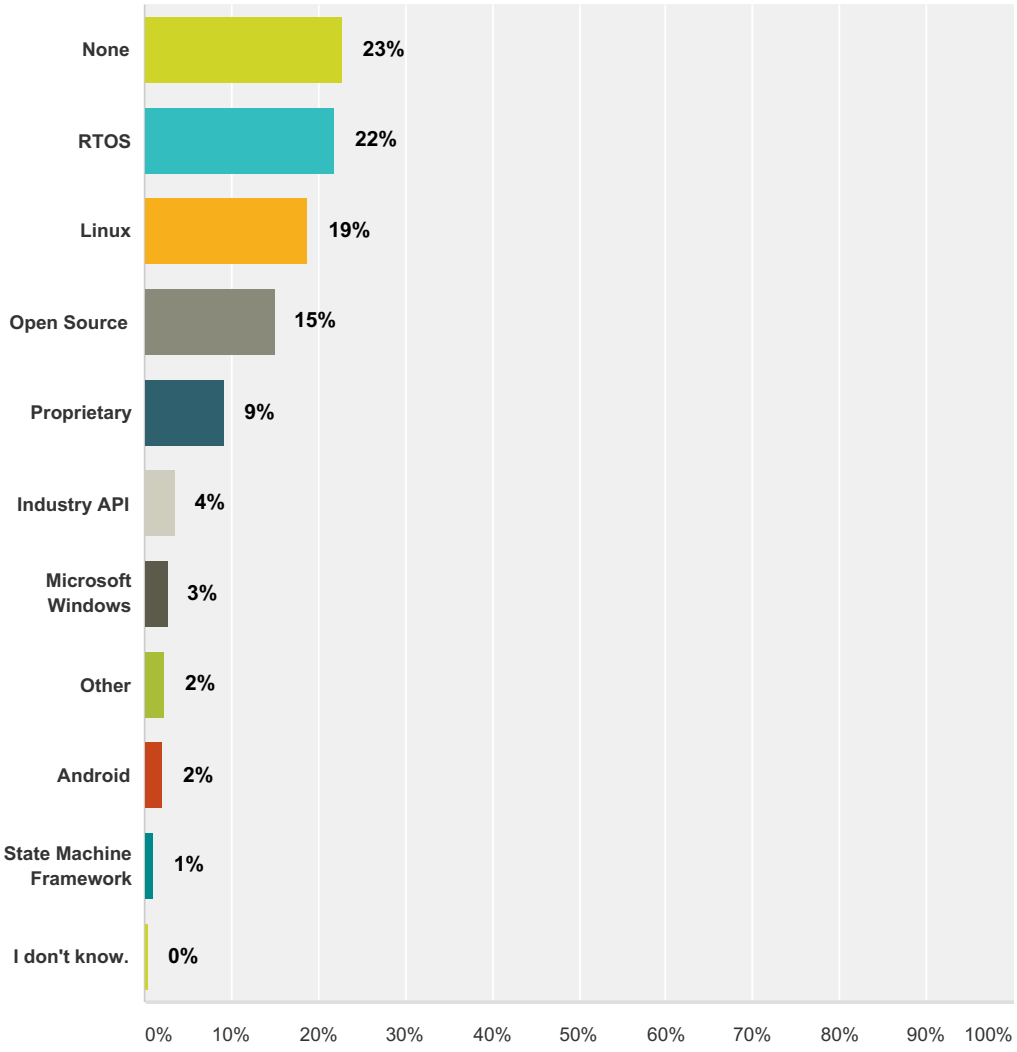
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
None	0%	0
1 person	21%	355
2-4 people	48%	820
5-9 people	17%	297
10-19 people	6%	108
20+ people	8%	132
I don't know.	1%	14
Total		1,726

Q7 What type of primary operating system do you expect to run on the main processor?

Answered: 1,726 Skipped: 0



Answer Choices	Responses	
None	23%	394
RTOS	22%	379
Linux	19%	326
Open Source	15%	258
Proprietary	9%	158
Industry API	4%	61
Microsoft Windows	3%	48
Other	2%	41

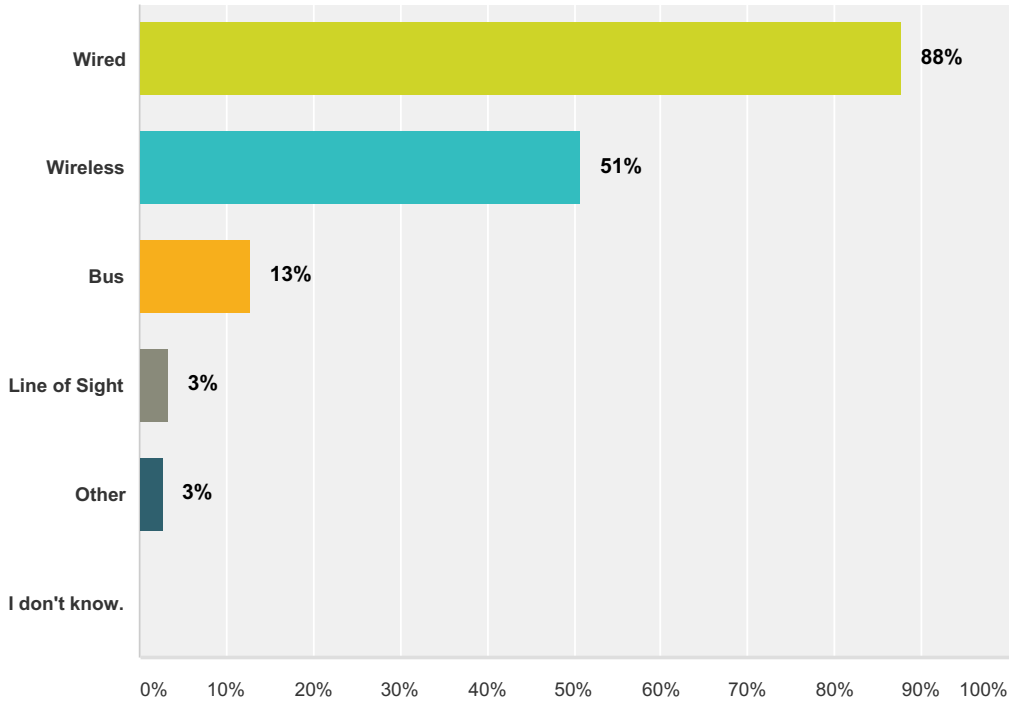
Android	**2%**	35
State Machine Framework	**1%**	18
I don't know.	**0%**	8
Total		**1,726**

#	Other (please specify)	Date
1	IBM Rhapsody	1/30/2017 10:51 PM
2	PLC	1/28/2017 3:29 AM
3	4e4th / Forth	1/28/2017 12:34 AM
4	No main processor.	1/25/2017 11:38 PM
5	x	1/25/2017 8:31 PM
6	I wrote my own RTOS and am using it	1/25/2017 7:35 PM
7	TI	1/25/2017 6:20 PM
8	IBM System Z embedded OS	1/25/2017 4:59 PM
9	Proprietary state machines for MCU, linux for CPU	1/25/2017 4:33 PM
10	Forth	1/22/2017 11:30 AM
11	MicroEJ	1/21/2017 8:30 PM
12	iTRON	1/20/2017 4:10 PM
13	customer dependant	1/19/2017 1:22 PM
14	ThreadX	1/19/2017 4:34 AM
15	based on quantum leaps framework	1/19/2017 12:46 AM
16	Linux	1/18/2017 6:12 PM
17	Win Embedded compact	1/18/2017 5:21 PM
18	bare metal application	1/18/2017 5:15 PM
19	bare metal python	1/17/2017 9:54 PM
20	bare metal	1/16/2017 11:07 PM
21	MicroEJ OS	1/16/2017 10:10 PM
22	while(1)	1/16/2017 5:36 AM
23	SafeRTOS	1/14/2017 9:24 AM
24	Dependig on the Electronics System Type	1/13/2017 9:31 AM
25	It will be Forth-based, although you may or may not consider that an OS.	1/13/2017 5:34 AM
26	Contiki	1/12/2017 7:57 PM
27	commercial RTOS + Linux + Android	1/12/2017 6:26 PM
28	Arduino IDE	1/12/2017 4:51 PM
29	Xenomai 2.6+	1/12/2017 4:45 PM
30	Multiple, linux, android and proprietary	1/12/2017 4:40 PM
31	Linux now, FreeRTOS and Windows 10 in the works	1/12/2017 4:39 PM
32	Windows CE	1/12/2017 4:37 PM
33	Chip vendor RTOS and QP	1/12/2017 4:27 PM
34	In house custom linux kernel	1/12/2017 4:21 PM

35	Linux / QNX	1/12/2017 4:21 PM
36	MQX and ThreadX	1/12/2017 4:14 PM
37	No OS, a priority event queue system	1/12/2017 4:05 PM
38	RTEMS	1/12/2017 4:05 PM
39	mbed os	1/12/2017 2:02 AM
40	FreeDOS	1/11/2017 11:40 PM
41	this is a multi core device that will be running linux and mqx	1/10/2017 7:02 PM

Q8 If you know, what types of external connections will your current project have? (select all that apply)

Answered: 1,717 Skipped: 9



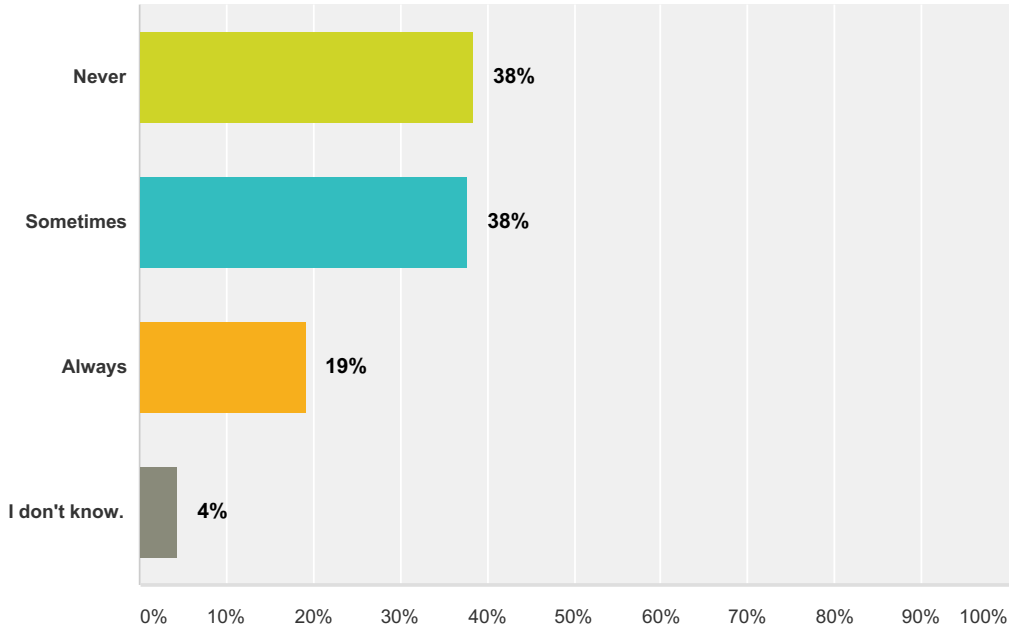
Answer Choices	Responses	
Wired	88%	1,507
Wireless	51%	872
Bus	13%	218
Line of Sight	3%	59
Other	3%	45
I don't know.	0%	0
Total Respondents: 1,717		

#	Other (please specify)	Date
1	none	1/31/2017 4:04 PM
2	USB	1/28/2017 2:30 PM
3	x	1/25/2017 8:31 PM
4	IR bidirectional - sorta like TV clicker but IR both ways	1/25/2017 7:35 PM
5	FPGA	1/25/2017 4:52 PM
6	We	1/25/2017 4:36 PM
7	Relay contacts.	1/25/2017 4:32 PM

8	4-20mA	1/23/2017 3:12 PM
9	analog signals out	1/20/2017 7:39 PM
10	lvds	1/20/2017 4:01 AM
11	HDMI	1/19/2017 8:27 PM
12	LORA, 802.15.4, HPGP	1/19/2017 8:25 PM
13	customer dependant	1/19/2017 1:22 PM
14	serial lines, proprietary buses	1/19/2017 10:02 AM
15	none	1/19/2017 3:33 AM
16	fiber GPON	1/18/2017 5:41 PM
17	HDBaseT	1/18/2017 5:34 PM
18	Synchronous communication port (e.g. HDLC)	1/17/2017 8:37 PM
19	SRIO	1/17/2017 9:30 AM
20	1 bluetooth, two wired	1/17/2017 1:57 AM
21	Analog camera	1/16/2017 10:21 PM
22	Plugs into outlet	1/16/2017 6:46 PM
23	4 analog inputs, 2 MosFets out.	1/16/2017 5:50 PM
24	I/O control to servo(s)	1/15/2017 4:15 AM
25	None	1/15/2017 12:38 AM
26	RS485	1/14/2017 8:27 AM
27	Reflective Memory	1/14/2017 1:57 AM
28	fire line	1/13/2017 11:05 PM
29	I'm hoping for a touch-screen but if I can't get that, I'll use a mouse or touchpad or trackball...	1/13/2017 5:34 AM
30	Power supply input / output (it's a power supply!)	1/13/2017 4:08 AM
31	ARINC429	1/13/2017 12:06 AM
32	Custom gigabit fiber connections & FPGA based routing network.	1/12/2017 6:11 PM
33	Ethernet TCP	1/12/2017 5:54 PM
34	no external network connections	1/12/2017 5:45 PM
35	None	1/12/2017 5:43 PM
36	SD card	1/12/2017 5:40 PM
37	JTAG	1/12/2017 5:07 PM
38	I2C, flash storage	1/12/2017 4:39 PM
39	no communication connections	1/12/2017 4:39 PM
40	SpaceWire	1/12/2017 4:05 PM
41	n/a	1/11/2017 9:09 PM
42	none	1/11/2017 11:03 AM
43	Sub-1Ghz wireless connections	1/10/2017 7:02 PM
44	None	1/10/2017 7:00 PM
45	audio and analog voltage I/O	1/10/2017 6:56 PM

Q9 When, if at all, will your current project be connected (directly or indirectly) to the Internet?

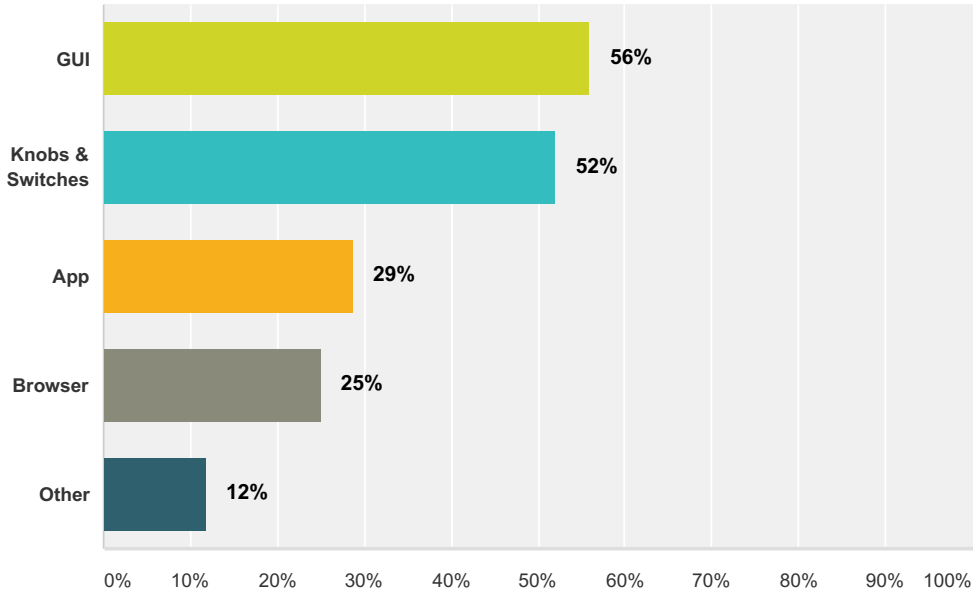
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
Never	38%	664
Sometimes	38%	653
Always	19%	333
I don't know.	4%	76
Total		1,726

Q10 If you know, how will users interact with your current project? (select all that apply)

Answered: 1,698 Skipped: 28



Answer Choices	Responses	
GUI	**56%**	951
Knobs & Switches	**52%**	882
App	**29%**	488
Browser	**25%**	427
Other	**12%**	201
Total Respondents: 1,698		

#	Other (please specify)	Date
1	a combination of GUI/buttons/web interface	2/2/2017 6:42 PM
2	RFID	2/1/2017 10:05 PM
3	Voice Command	2/1/2017 11:31 AM
4	Via a Windows application	2/1/2017 10:43 AM
5	Serial RS232	1/31/2017 6:34 PM
6	MODBUS/RTU or MODBUS/TCP	1/31/2017 6:01 PM
7	Command line	1/31/2017 5:48 PM
8	IO-Link industrial protocol	1/31/2017 2:14 PM
9	via PC memory storage system - product is SSD drive.	1/30/2017 9:46 PM
10	SMS	1/30/2017 8:03 AM
11	linux like CLI	1/29/2017 9:58 PM

12	PC application	1/29/2017 7:03 PM
13	SCADA	1/29/2017 5:23 AM
14	custom remote control panel	1/27/2017 5:30 AM
15	proprietary tool	1/26/2017 5:55 PM
16	touch screen	1/26/2017 1:54 PM
17	Trough car switchen, then trough a Body Computer Module (BCM)	1/26/2017 9:08 AM
18	Device periodically transmits data to back end database, then user can view that data in Web Browser.	1/26/2017 1:17 AM
19	custom software	1/26/2017 12:01 AM
20	API	1/25/2017 9:21 PM
21	By a custom RF display	1/25/2017 9:10 PM
22	Device controlled with Corba calls and REST API	1/25/2017 9:05 PM
23	computer to computer control also included.	1/25/2017 8:53 PM
24	x	1/25/2017 8:31 PM
25	See #8 IR com for one-time setup then motion & illumination i.e. No user I/F after init setup	1/25/2017 7:35 PM
26	It will be via web browser, but the server is in the cloud. Our IoT gateway is responsible for sending the nodes data to this central server.	1/25/2017 7:13 PM
27	Wireless	1/25/2017 6:20 PM
28	via network or USB	1/25/2017 5:40 PM
29	na	1/25/2017 5:07 PM
30	Voice	1/25/2017 4:34 PM
31	Via web browser, via the cloud.	1/25/2017 4:33 PM
32	In-band SES, serial command/debug port	1/25/2017 4:30 PM
33	Command line interface	1/24/2017 3:03 PM
34	OEM, controller read our output	1/23/2017 3:12 PM
35	via build-in usb<->serial converter or telnet	1/22/2017 12:27 AM
36	indirect via phys. IF (e.g. PCIe, USB, Ethernet)	1/20/2017 6:00 PM
37	via host controller	1/20/2017 2:10 PM
38	No interaction	1/20/2017 10:39 AM
39	pc tools	1/20/2017 7:23 AM
40	Command Line Interface	1/20/2017 5:12 AM
41	none	1/19/2017 4:43 PM
42	SCADA System	1/19/2017 4:39 PM
43	device that os interacts with	1/19/2017 3:07 PM
44	Cloud	1/19/2017 1:22 PM
45	Touch screen	1/19/2017 9:36 AM
46	console port	1/19/2017 5:02 AM
47	buttons	1/19/2017 3:33 AM
48	Via CUI	1/19/2017 1:23 AM
49	voice	1/19/2017 12:47 AM
50	Current project is headless	1/19/2017 12:06 AM
51	multiple interfaces	1/18/2017 9:49 PM

52	3	1/18/2017 7:34 PM
53	command-line utilities	1/18/2017 7:04 PM
54	system automation connections	1/18/2017 7:03 PM
55	via a cloud service	1/18/2017 6:36 PM
56	Voice, Barcode	1/18/2017 6:04 PM
57	autonomous, headless system	1/18/2017 6:00 PM
58	MIB interface to host system	1/18/2017 5:41 PM
59	Fieldbus	1/18/2017 5:37 PM
60	IDE	1/18/2017 5:30 PM
61	embedded in a larger system.	1/18/2017 5:15 PM
62	Via fingerprint sensor.	1/18/2017 5:15 PM
63	Users will not directly interact with our product. Our product will interact with sensors and actuators in vehicles.	1/18/2017 5:14 PM
64	No user interaction	1/18/2017 5:13 PM
65	CAN, RS-485	1/18/2017 3:43 PM
66	Element Management System and Network Management System	1/18/2017 3:34 PM
67	not at all	1/18/2017 8:33 AM
68	Serial console	1/17/2017 10:04 PM
69	telnet-like interface	1/17/2017 8:37 PM
70	Fieldbus	1/17/2017 3:38 PM
71	The user cannot directly interact with the devices. The user will only be able to view data through a web portal	1/17/2017 2:07 PM
72	via PC based tool	1/17/2017 10:14 AM
73	Bespoke PC App & Ethernet	1/17/2017 9:44 AM
74	Via the cloud	1/17/2017 9:39 AM
75	Terminal	1/17/2017 8:45 AM
76	Proprietary communications protocol over TLS connection.	1/17/2017 4:58 AM
77	1 app, 1 GUI, 1 non-GUI front panel	1/17/2017 1:57 AM
78	Central (non-embedded) web portal	1/17/2017 1:12 AM
79	LED and keypad	1/16/2017 10:13 PM
80	Remote control by Leshan	1/16/2017 10:10 PM
81	laser / SMR	1/16/2017 9:50 PM
82	Indirectly via web browser (Internet-PC-USB_rf-System)	1/16/2017 9:10 PM
83	depends on customer's integration. ssh is also a primary method.	1/16/2017 8:17 PM
84	it is a usb device	1/16/2017 8:06 PM
85	They won't, it's part of the basic structure of an end product	1/16/2017 7:32 PM
86	PC tool	1/16/2017 7:32 PM
87	EIA-232	1/16/2017 6:56 PM
88	primarily through a command-line interface	1/16/2017 6:50 PM
89	command line	1/16/2017 6:31 PM
90	Indirectly via automobile headunit over a CAN bus to our amplifier	1/16/2017 6:16 PM
91	Text terminal	1/16/2017 6:15 PM
92	via an external server	1/16/2017 1:41 PM

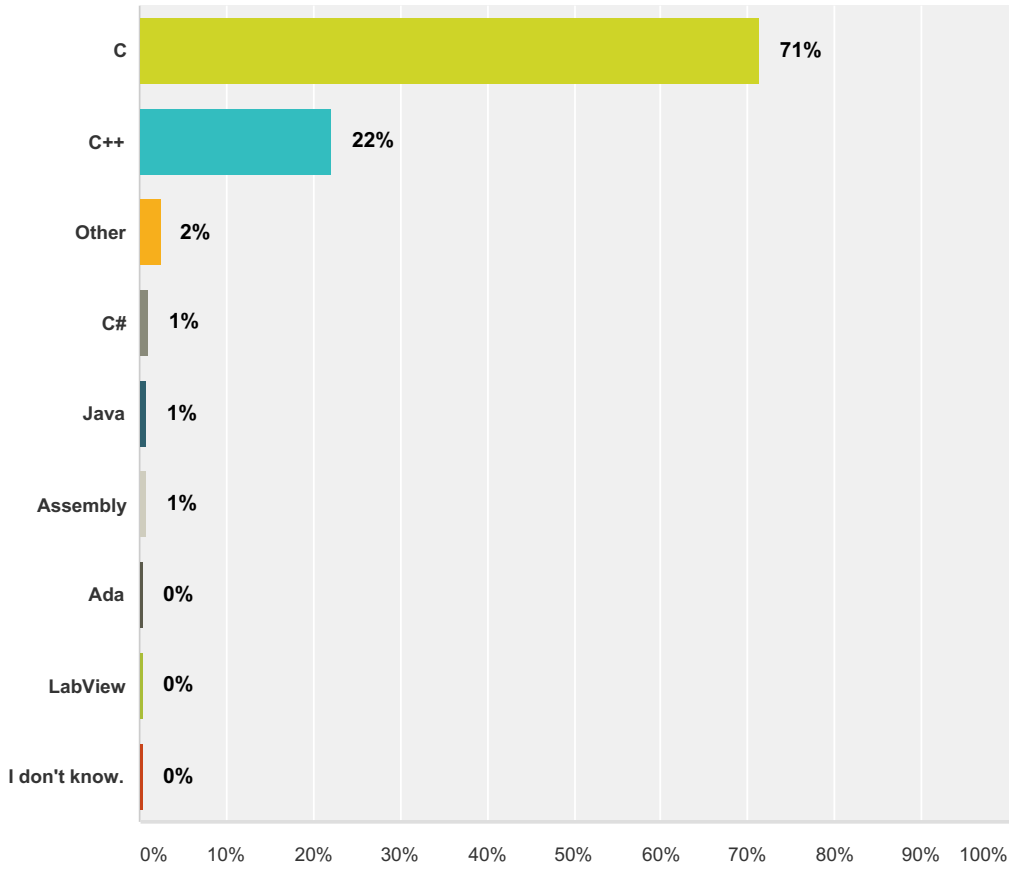
93	Ssh	1/16/2017 1:32 PM
94	custom 2.4ghz Remotecontrol	1/16/2017 8:03 AM
95	PC based	1/16/2017 5:39 AM
96	Development tools framework (I build microprocessors)	1/16/2017 5:36 AM
97	Via proprietary Server and Desktop software	1/16/2017 3:38 AM
98	PC user interface (FDT DTM)	1/15/2017 5:34 PM
99	(TBD) Web browser on internal LAN	1/15/2017 4:15 AM
100	Via an app control computer	1/15/2017 2:48 AM
101	Web application	1/14/2017 10:12 PM
102	Serial comms (command line)	1/14/2017 6:55 PM
103	No user interaction	1/14/2017 9:40 AM
104	Command line	1/14/2017 1:57 AM
105	Via an application running on a PC connected via USB	1/13/2017 4:19 PM
106	via CLI	1/13/2017 4:11 PM
107	Connected to industrial control system / plant automation	1/13/2017 1:04 PM
108	communication via diverse CANopen-Tools	1/13/2017 12:06 PM
109	CAN open protocol	1/13/2017 11:28 AM
110	mud pulse transmittion channel	1/13/2017 10:20 AM
111	It is an autonomous data collection unit in a nano satellite; data transfer over RF interface at regular intervals	1/13/2017 8:46 AM
112	via another system UI	1/13/2017 8:09 AM
113	Engine Controller development tools and reference designs	1/13/2017 7:07 AM
114	cli	1/13/2017 7:01 AM
115	networking protocols	1/13/2017 6:34 AM
116	interactive test tool; APIs over vehicle buses	1/13/2017 6:27 AM
117	Conversational UI	1/13/2017 5:46 AM
118	via touch screen or mouse/touchpad/trackball	1/13/2017 5:34 AM
119	device drivers (deeply embedded)	1/13/2017 4:26 AM
120	via system comms interface	1/13/2017 4:08 AM
121	i2c	1/13/2017 2:15 AM
122	Flight control system	1/13/2017 12:06 AM
123	Audio, video	1/12/2017 10:44 PM
124	MEMS sensors	1/12/2017 10:31 PM
125	Text to speech	1/12/2017 10:26 PM
126	Desktop application (Python) for configuration and data analysis.	1/12/2017 10:22 PM
127	dedicated host master	1/12/2017 10:22 PM
128	ssh	1/12/2017 9:11 PM
129	SSH/NETCONF	1/12/2017 9:08 PM
130	EtherCAT	1/12/2017 9:08 PM
131	Attached medical device	1/12/2017 8:11 PM
132	No interaction needed - much like a pre-configured router.	1/12/2017 8:04 PM
133	Api over uart, bluetooth or ip	1/12/2017 7:36 PM

134	Interface device that is never "seen" by the end user.	1/12/2017 6:59 PM
135	Cots Device connected to the product	1/12/2017 6:47 PM
136	USB serial port	1/12/2017 6:37 PM
137	No interaction once installed	1/12/2017 6:36 PM
138	IR	1/12/2017 6:30 PM
139	Rest API CLI tool as well	1/12/2017 6:27 PM
140	Product communicate with machine not user	1/12/2017 5:52 PM
141	via web server connection to the embedded device	1/12/2017 5:47 PM
142	M2M - via connected components	1/12/2017 5:45 PM
143	OS driver	1/12/2017 5:41 PM
144	aircraft cockpit display	1/12/2017 5:41 PM
145	Via cloud server	1/12/2017 5:37 PM
146	Via a GUI on a PC, who's application communicates with the project/product/tester via Ethernet.	1/12/2017 5:28 PM
147	AWS Voice services, Microsoft backend services	1/12/2017 5:19 PM
148	No interaction.	1/12/2017 5:18 PM
149	serial, LED indicators, selection jumpers	1/12/2017 5:17 PM
150	SNMP, RS-232/485	1/12/2017 5:16 PM
151	Command line interface initially, then GUI	1/12/2017 5:14 PM
152	Another processor in the system provides human interface	1/12/2017 5:10 PM
153	physical touchscreen HMI	1/12/2017 5:09 PM
154	Remote control over the internet.	1/12/2017 5:04 PM
155	Not sure	1/12/2017 5:03 PM
156	It's ECU in car	1/12/2017 5:02 PM
157	through SPI interface.	1/12/2017 5:01 PM
158	Are subsystems used by a master via I2C & WLAN	1/12/2017 4:57 PM
159	Windows based GUI	1/12/2017 4:53 PM
160	PLC	1/12/2017 4:50 PM
161	This project routes data to a server that routes data to another server. Most interactions will be with that data, so interactions will be indirect or limited.	1/12/2017 4:45 PM
162	autonomous system	1/12/2017 4:43 PM
163	No user interaction	1/12/2017 4:41 PM
164	command line configuration + through a Network Management System (using a GUI)	1/12/2017 4:38 PM
165	audio only	1/12/2017 4:35 PM
166	automotive specific networks	1/12/2017 4:33 PM
167	Via a web portal in cloud, that talks with embedded device	1/12/2017 4:31 PM
168	cloud controlled	1/12/2017 4:26 PM
169	Purpose built rugged handheld computer devices, Windows PCs, and manually loading/unloading data from an SD card.	1/12/2017 4:26 PM
170	Online IOT interface	1/12/2017 4:25 PM
171	Serial String Commands	1/12/2017 4:25 PM
172	CLI	1/12/2017 4:21 PM

173	Voice control	1/12/2017 4:20 PM
174	Via a proprietary text display / keypad.	1/12/2017 4:18 PM
175	Custom defense system bus	1/12/2017 4:16 PM
176	Via standard protocol interface tools	1/12/2017 4:12 PM
177	CLI over SSH	1/12/2017 4:09 PM
178	Remotely accessed TUI (Textual User Interface) - text menus	1/12/2017 4:08 PM
179	via RS232 communication from a PC running Matlab/C#	1/12/2017 4:05 PM
180	TM/TC from ground station	1/12/2017 4:05 PM
181	command line and voice are possibilities	1/12/2017 8:39 AM
182	Voice	1/12/2017 5:12 AM
183	Via an API	1/12/2017 2:34 AM
184	Com port Terminal	1/11/2017 7:41 PM
185	Ethernet services	1/11/2017 1:04 PM
186	via a setup program	1/11/2017 12:37 PM
187	api	1/11/2017 12:07 PM
188	serial console for configuration only	1/11/2017 10:57 AM
189	Telnet, text interface (serial)	1/11/2017 10:27 AM
190	Satellite ground station	1/11/2017 10:08 AM
191	in-house application/service	1/11/2017 9:46 AM
192	Smart card	1/11/2017 9:43 AM
193	remote API over Ethernet	1/11/2017 9:40 AM
194	industrial controllers	1/10/2017 7:41 PM
195	Physical accelerometers, light detection, sound	1/10/2017 7:39 PM
196	Project is a software library with no specified UI	1/10/2017 7:38 PM
197	C# API	1/10/2017 7:18 PM
198	Headend meter data management software	1/10/2017 7:09 PM
199	depends on end product	1/10/2017 7:02 PM
200	UART	1/10/2017 7:01 PM
201	custom application running on VxWorks	1/10/2017 6:56 PM

Q11 What is the primary programming language for your current project?

Answered: 1,726 Skipped: 0



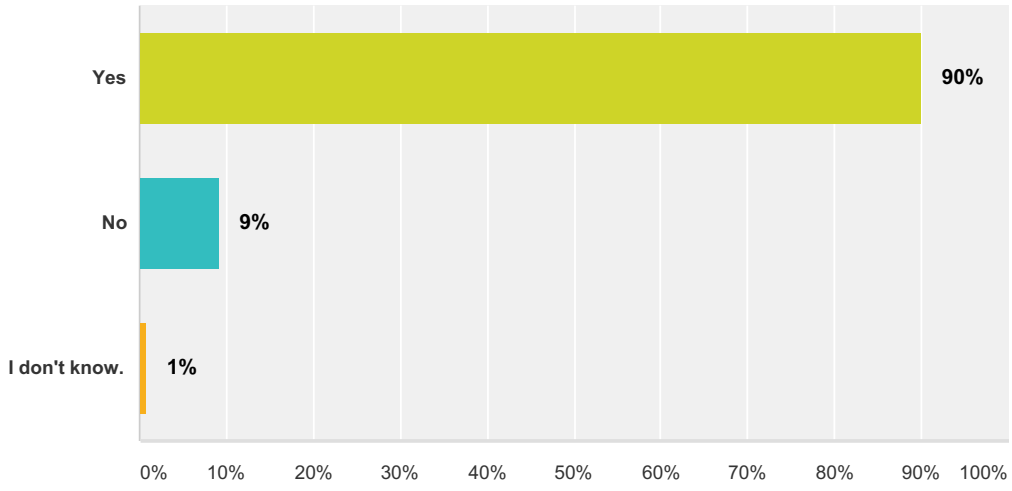
Answer Choices	Responses	
C	**71%**	1,234
C++	**22%**	381
Other	**2%**	42
C#	**1%**	17
Java	**1%**	16
Assembly	**1%**	15
Ada	**0%**	8
LabView	**0%**	7
I don't know.	**0%**	6
Total		**1,726**

#	Other (please specify)	Date
1	Python, shell	1/31/2017 3:30 AM

2	PLC	1/28/2017 3:30 AM
3	Forth	1/28/2017 12:37 AM
4	Multiple	1/26/2017 6:05 PM
5	C and possibly FORTH	1/25/2017 11:40 PM
6	Visual Basic	1/25/2017 4:34 PM
7	Forth	1/22/2017 11:32 AM
8	mix of C and C++ (reuse of existing source code)	1/20/2017 9:03 AM
9	C and Python also Labview	1/19/2017 2:03 PM
10	Fairmount Automation DesignPad	1/19/2017 1:49 PM
11	android	1/18/2017 10:54 PM
12	Python	1/18/2017 7:05 PM
13	Rather not say at this time.	1/18/2017 5:36 PM
14	MBSD w auto-generated c code	1/18/2017 5:31 PM
15	vhdl	1/18/2017 3:56 PM
16	python	1/17/2017 9:56 PM
17	1 C, 1 Java, 1 C#	1/17/2017 1:58 AM
18	python	1/17/2017 1:28 AM
19	Javascript	1/16/2017 11:09 PM
20	VB.net	1/16/2017 8:22 PM
21	Simulink	1/15/2017 12:11 AM
22	IEC61131	1/13/2017 4:48 PM
23	Simulink/MBD to C	1/13/2017 2:33 PM
24	Forth, with some assembly low-level support.	1/13/2017 5:36 AM
25	Python	1/12/2017 9:02 PM
26	Forth	1/12/2017 5:50 PM
27	Forth	1/12/2017 5:34 PM
28	Matlab stateflow and simulink	1/12/2017 5:06 PM
29	Python	1/12/2017 4:58 PM
30	python	1/12/2017 4:42 PM
31	Go	1/12/2017 4:31 PM
32	C and C++	1/12/2017 4:28 PM
33	python	1/12/2017 4:23 PM
34	modell-based development	1/12/2017 4:23 PM
35	Python	1/12/2017 4:20 PM
36	Python	1/12/2017 4:14 PM
37	javascript	1/12/2017 4:13 PM
38	js	1/12/2017 4:11 PM
39	Python, Javascript	1/12/2017 4:08 PM
40	Lua	1/12/2017 4:08 PM
41	Python	1/11/2017 9:04 PM
42	Python	1/10/2017 7:42 PM

Q12 Does your team use a version control system?

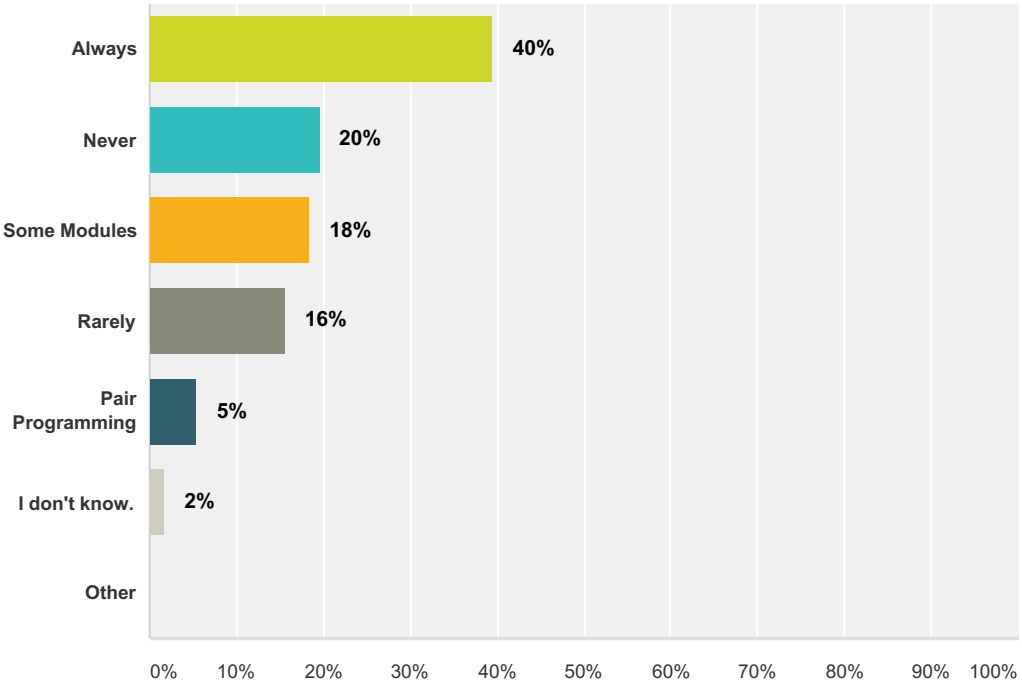
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
Yes	**90%**	1,553
No	**9%**	158
I don't know.	**1%**	15
Total		**1,726**

Q13 Are peer source code reviews a part of the software development process?

Answered: 1,726 Skipped: 0

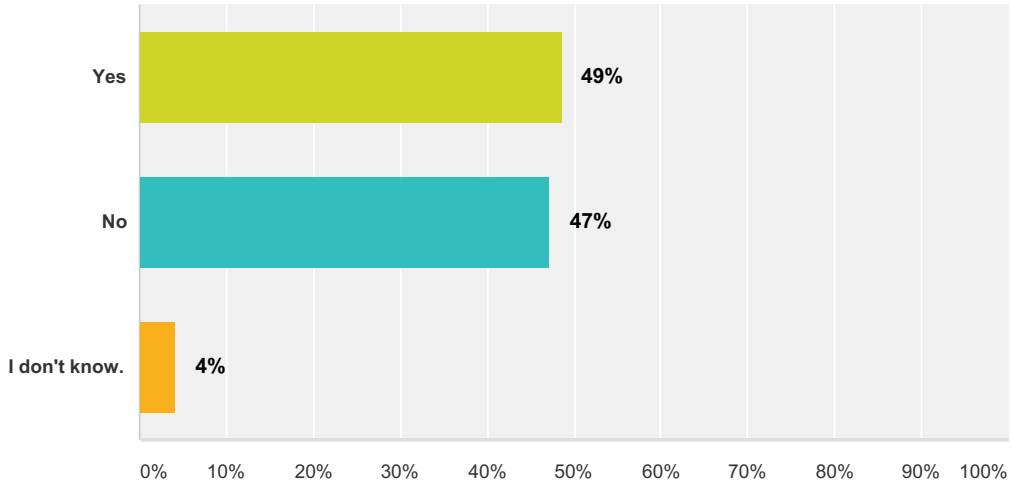


Answer Choices	Responses	
Always	40%	682
Never	20%	337
Some Modules	18%	316
Rarely	16%	269
Pair Programming	5%	94
I don't know.	2%	28
Other	0%	0
Total		1,726

#	Other (please specify)	Date
	There are no responses.	

Q14 Is source code run through one or more static analysis tools (e.g., PC/Lint or Coverity)?

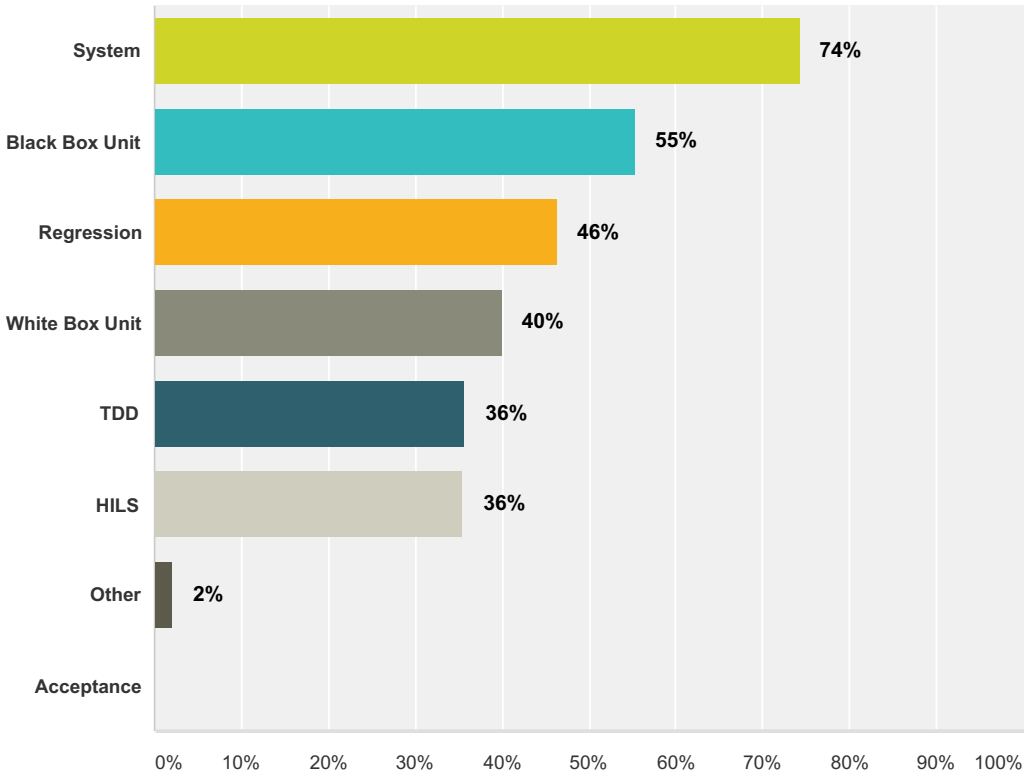
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
Yes	**49%**	840
No	**47%**	813
I don't know.	**4%**	73
Total		**1,726**

Q15 If you know, what kinds of testing will be performed? (select all that apply)

Answered: 1,703 Skipped: 23



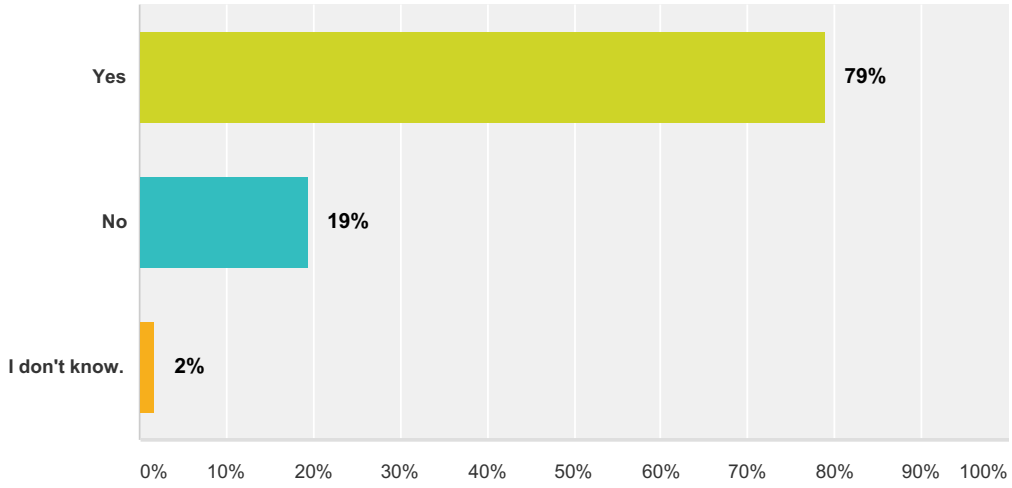
Answer Choices	Responses	
System	74%	1,267
Black Box Unit	55%	942
Regression	46%	788
White Box Unit	40%	684
TDD	36%	609
HILS	36%	605
Other	2%	37
Acceptance	0%	0
Total Respondents: 1,703		

#	Other (please specify)	Date
1	Manual Testing	1/31/2017 10:12 AM
2	Software-in-the-loop Testing	1/31/2017 3:39 AM
3	SW/SW integration testing	1/26/2017 5:56 PM
4	Lots and lots of in-field testing as this is research	1/25/2017 9:10 PM

5	x	1/25/2017 8:32 PM
6	used logic analyzer for IR code	1/25/2017 7:38 PM
7	Bed of Nails	1/25/2017 5:54 PM
8	almost all of the above	1/25/2017 5:00 PM
9	Performance testing and Characteristics verification	1/20/2017 5:14 AM
10	Manual testing with test descriptions for requirements	1/19/2017 3:48 PM
11	Continuous Integration	1/18/2017 9:11 PM
12	Right by design	1/18/2017 6:56 PM
13	There will be other testing mechanisms used.	1/18/2017 5:36 PM
14	Simulation	1/17/2017 9:40 AM
15	Customer acceptance testing	1/17/2017 9:39 AM
16	gues and check tuning	1/17/2017 1:28 AM
17	Don't know	1/16/2017 10:10 PM
18	Misra-c	1/16/2017 10:05 PM
19	Ad hoc (at best...)	1/16/2017 8:14 PM
20	we use 3 sigma testing, this means we test it 3 times and ship it (it is a prototype/development system)	1/16/2017 8:07 PM
21	testing against a requirements document	1/16/2017 5:56 PM
22	Simulation and verification	1/16/2017 5:37 AM
23	manual and automated integration testing; manual and automated system testing; in-vehicle testing; simulation; safety validation testing	1/13/2017 6:29 AM
24	Integration Testing	1/13/2017 5:54 AM
25	Not at the point where I'll decide that yet; I'm still trying to decide which hardware to use.	1/13/2017 5:36 AM
26	Software in Loop	1/13/2017 1:59 AM
27	utilize unit test framework and require unit tests for each functional component	1/12/2017 10:44 PM
28	requirements testing	1/12/2017 10:26 PM
29	FDA compliance testing	1/12/2017 10:04 PM
30	Software Unit Resilience Testing (active attempts at UUT destruction)	1/12/2017 5:50 PM
31	Squish for Qt components	1/12/2017 5:02 PM
32	Software-in-the-Loop Testing	1/12/2017 4:58 PM
33	Fuzzing	1/12/2017 4:21 PM
34	Integration testing, requirements-driven verification testing	1/12/2017 4:08 PM
35	Falt Injection	1/10/2017 7:40 PM
36	(This particular place lacks structure---I've been in aerospace)	1/10/2017 7:05 PM
37	V and V	1/10/2017 6:54 PM

Q16 Are known defects formally tracked (e.g., in a bug database or issue tracking system?

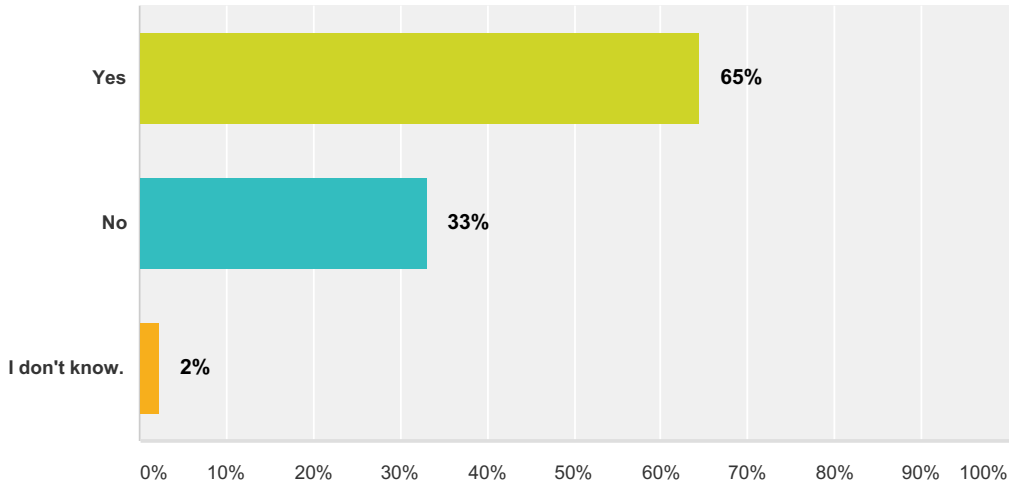
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
Yes	**79%**	1,362
No	**19%**	334
I don't know.	**2%**	30
Total		**1,726**

Q17 Is there a written coding standard in place that applies to your current project?

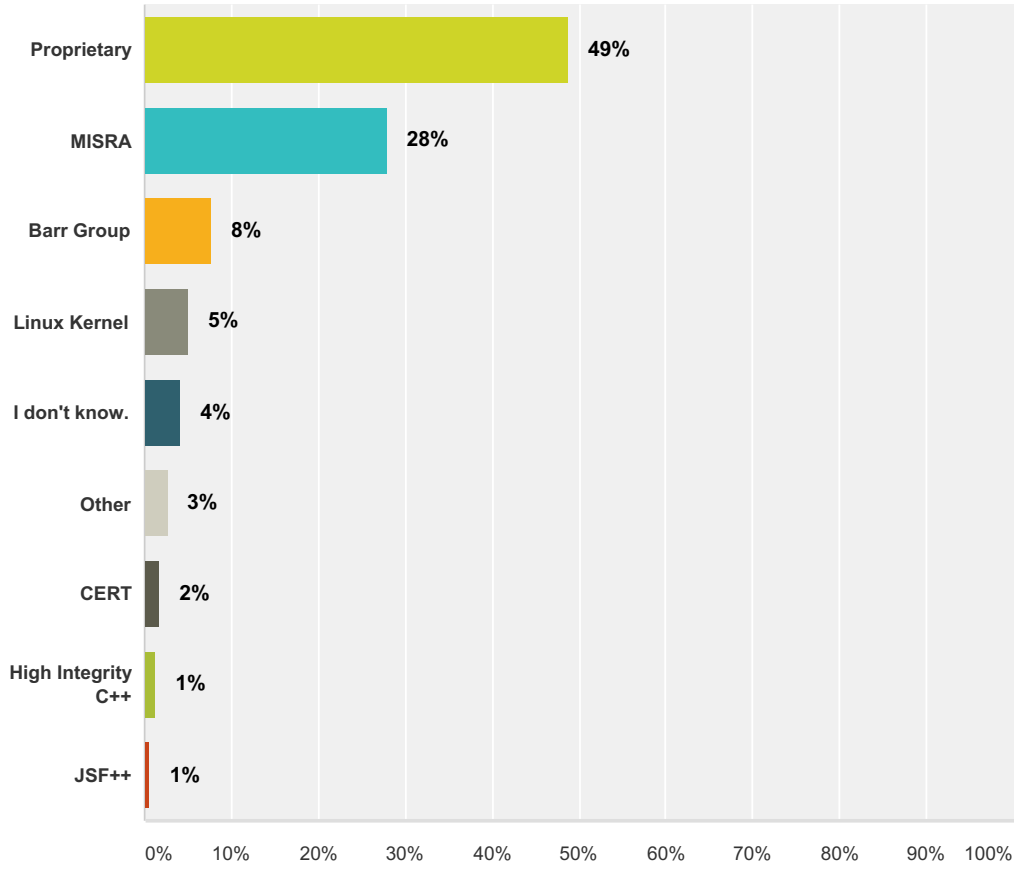
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
Yes	**65%**	1,115
No	**33%**	573
I don't know.	**2%**	38
Total		**1,726**

Q18 What is the primary basis of the coding standard that applies to your current project?

Answered: 1,115 Skipped: 611



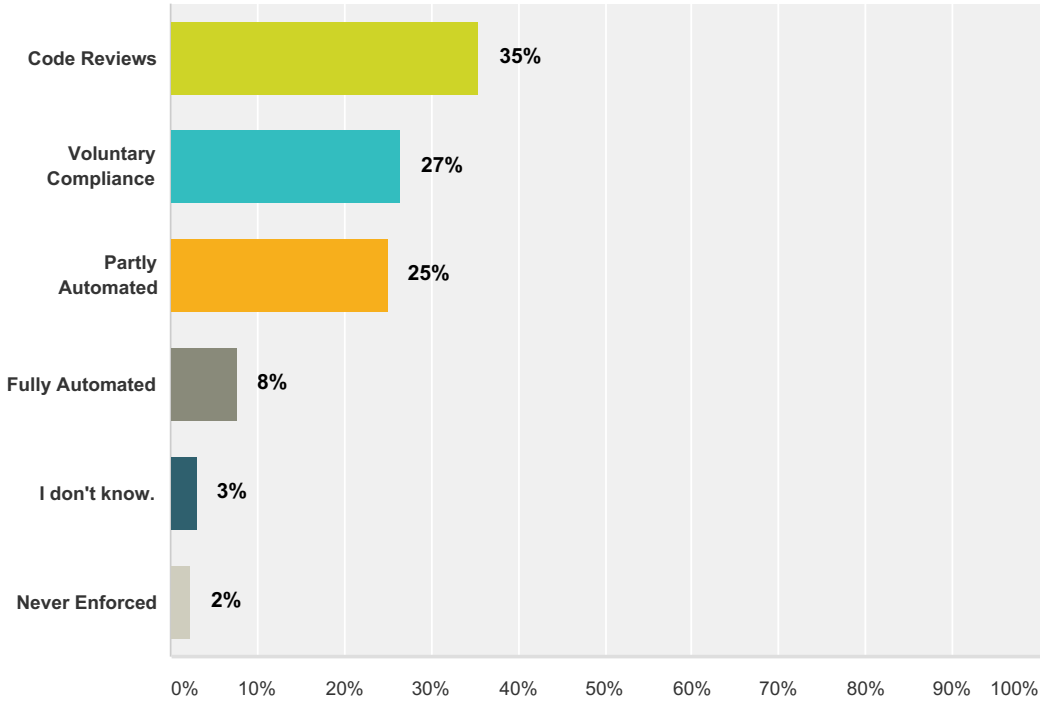
Answer Choices	Responses	
Proprietary	49%	545
MISRA	28%	312
Barr Group	8%	86
Linux Kernel	5%	55
I don't know.	4%	46
Other	3%	30
CERT	2%	19
High Integrity C++	1%	15
JSF++	1%	7
Total		1,115

#	Other (please specify)	Date

1	cannot say	1/26/2017 5:19 AM
2	Proprietary but now considering MISRA guidelines	1/25/2017 11:43 PM
3	x	1/25/2017 8:32 PM
4	Google coding standard	1/25/2017 4:38 PM
5	A less-strict policy derived from several sources.	1/25/2017 4:34 PM
6	Military nuclear propulsion standards	1/19/2017 1:50 PM
7	AS9100	1/18/2017 5:24 PM
8	misra-c plus in-house methodology for FDA review	1/16/2017 10:08 PM
9	IEC 62304	1/16/2017 8:47 PM
10	Google C++ Standard	1/14/2017 10:24 AM
11	combination	1/14/2017 3:10 AM
12	Qt's standard	1/14/2017 12:00 AM
13	Ganssle's with a little from Barr Group's	1/13/2017 1:21 PM
14	MISRA2012, CERT and Barr Group, in that order	1/13/2017 12:26 PM
15	Company standard ..a bit of everything	1/13/2017 11:47 AM
16	Misra and a proprietary coding standard	1/13/2017 7:10 AM
17	Micrium C Coding Standard	1/12/2017 11:38 PM
18	Michael Barr coding standards for C, Google C++ coding standards, and with Power-Of-Ten Standards	1/12/2017 10:19 PM
19	Contiki's standard	1/12/2017 7:59 PM
20	Google plus in-house modifications	1/12/2017 7:13 PM
21	Jack Ganssle	1/12/2017 6:17 PM
22	Parasoft/Ellemtel	1/12/2017 5:56 PM
23	This should be check all that apply	1/12/2017 5:23 PM
24	Started with JPL's flight safety critical standard and modified	1/12/2017 4:29 PM
25	MISRA for modell-based systems	1/12/2017 4:24 PM
26	Netrino	1/12/2017 4:14 PM
27	Internal	1/12/2017 4:12 PM
28	Company Internal	1/12/2017 4:08 PM
29	ECSS E40	1/12/2017 4:06 PM
30	PEP 8	1/11/2017 9:06 PM

Q19 How is the relevant coding standard enforced on your current project?

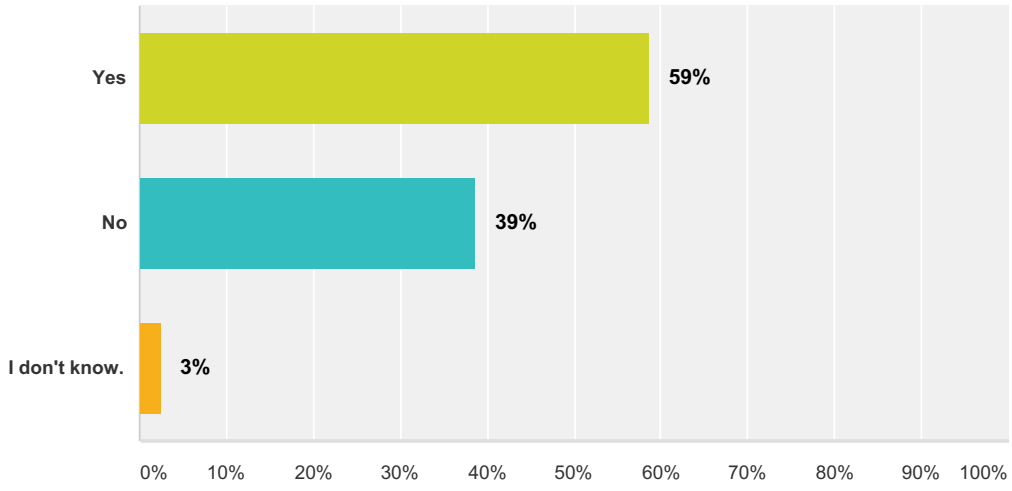
Answered: 1,115 Skipped: 611



Answer Choices	Responses	
Code Reviews	**35%**	395
Voluntary Compliance	**27%**	296
Partly Automated	**25%**	279
Fully Automated	**8%**	85
I don't know.	**3%**	34
Never Enforced	**2%**	26
Total		**1,115**

Q20 Is security one of the design considerations on your current project?

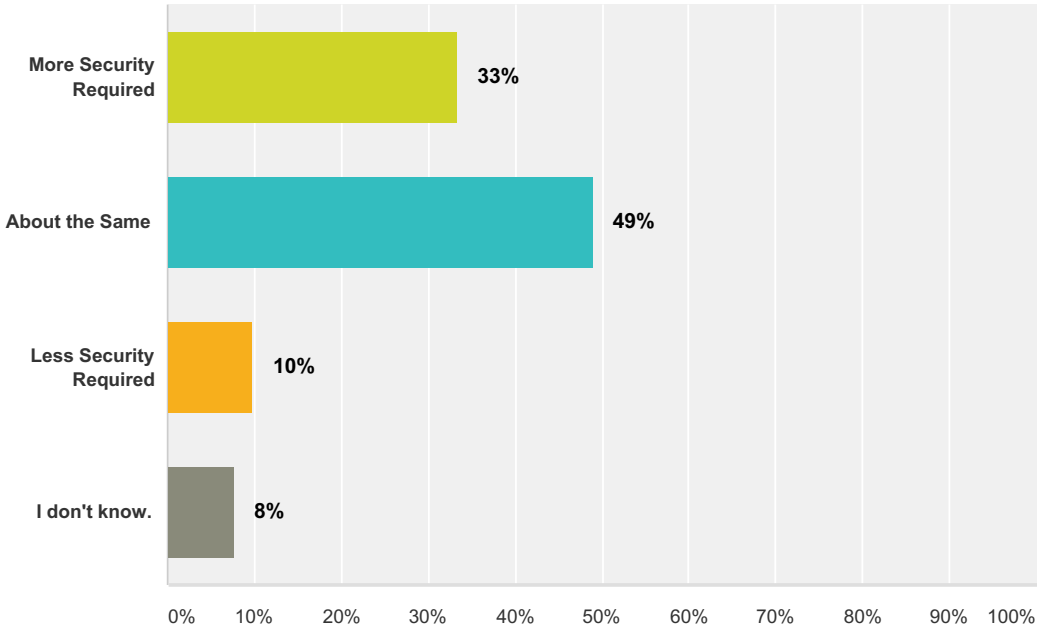
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
Yes	**59%**	1,014
No	**39%**	668
I don't know.	**3%**	44
Total		**1,726**

Q21 How do the security needs of your current project compare to your other recent projects?

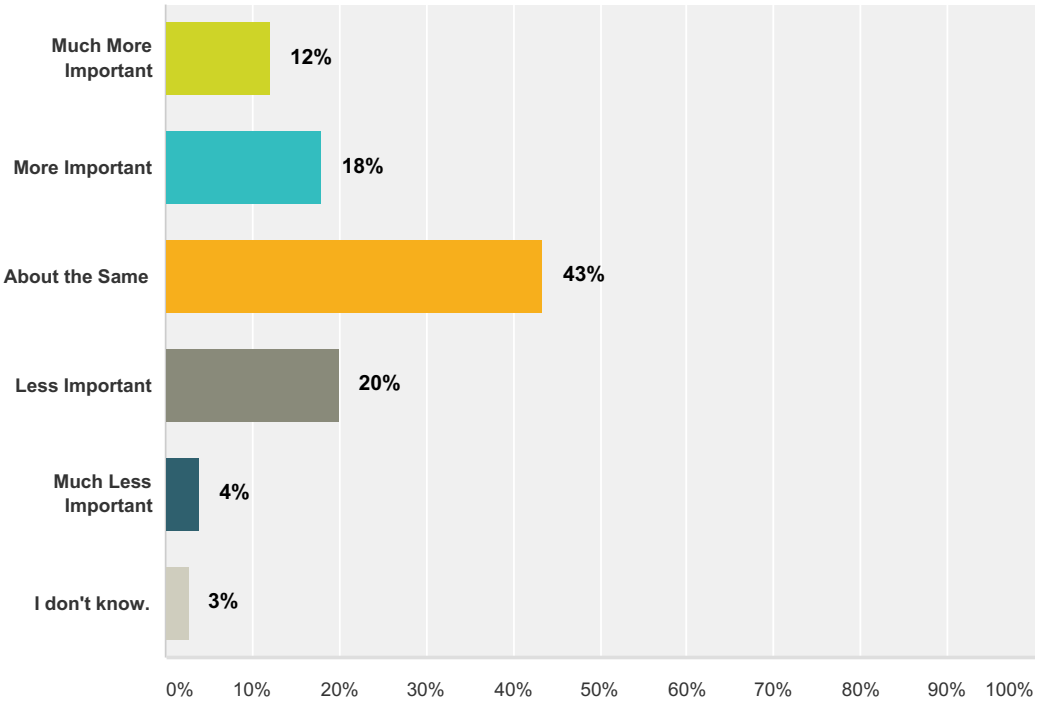
Answered: 1,717 Skipped: 9



Answer Choices	Responses	
More Security Required	**33%**	575
About the Same	**49%**	842
Less Security Required	**10%**	168
I don't know.	**8%**	132
Total		**1,717**

Q22 What is the relative importance of security vs. meeting the schedule on your current project?

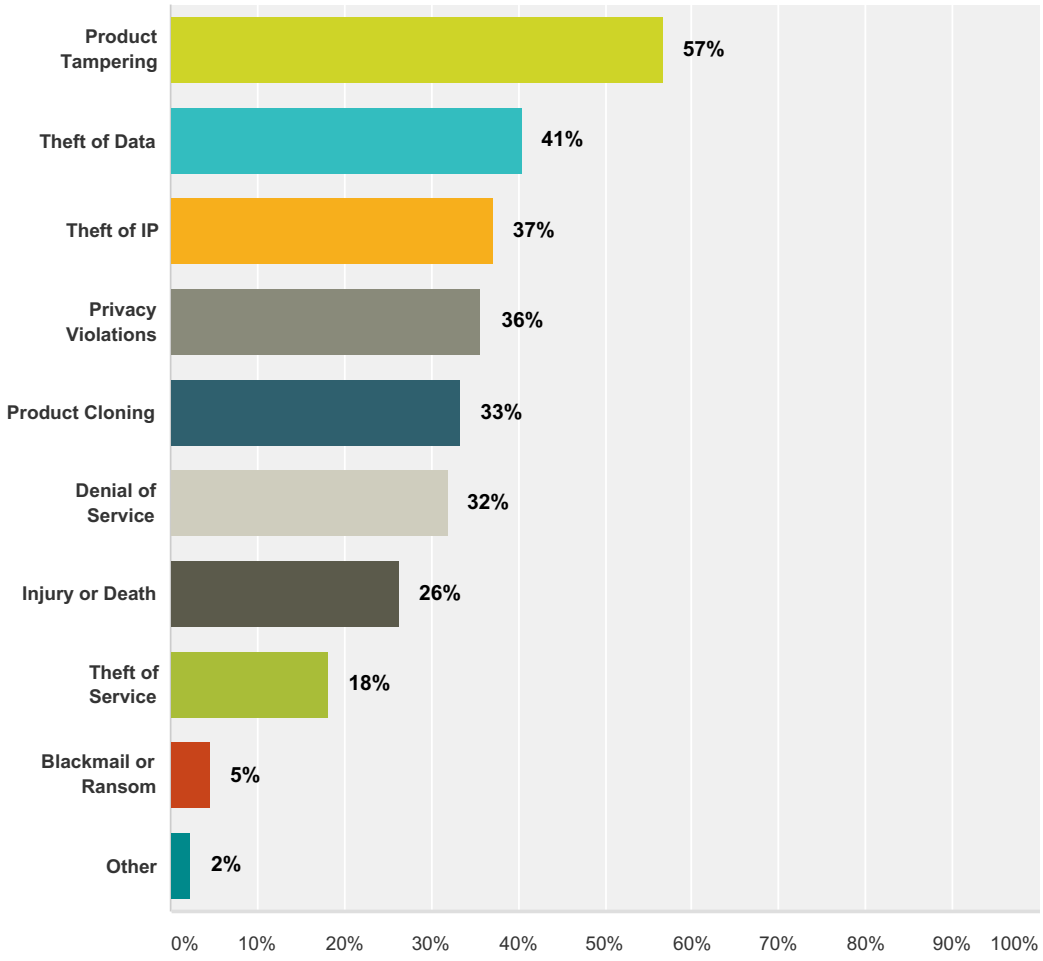
Answered: 1,014 Skipped: 712



Answer Choices	Responses	
Much More Important	**12%**	122
More Important	**18%**	182
About the Same	**43%**	440
Less Important	**20%**	203
Much Less Important	**4%**	40
I don't know.	**3%**	27
Total		**1,014**

Q23 If you know, what are the primary security concerns with your current project? (select all that apply)

Answered: 983 Skipped: 743



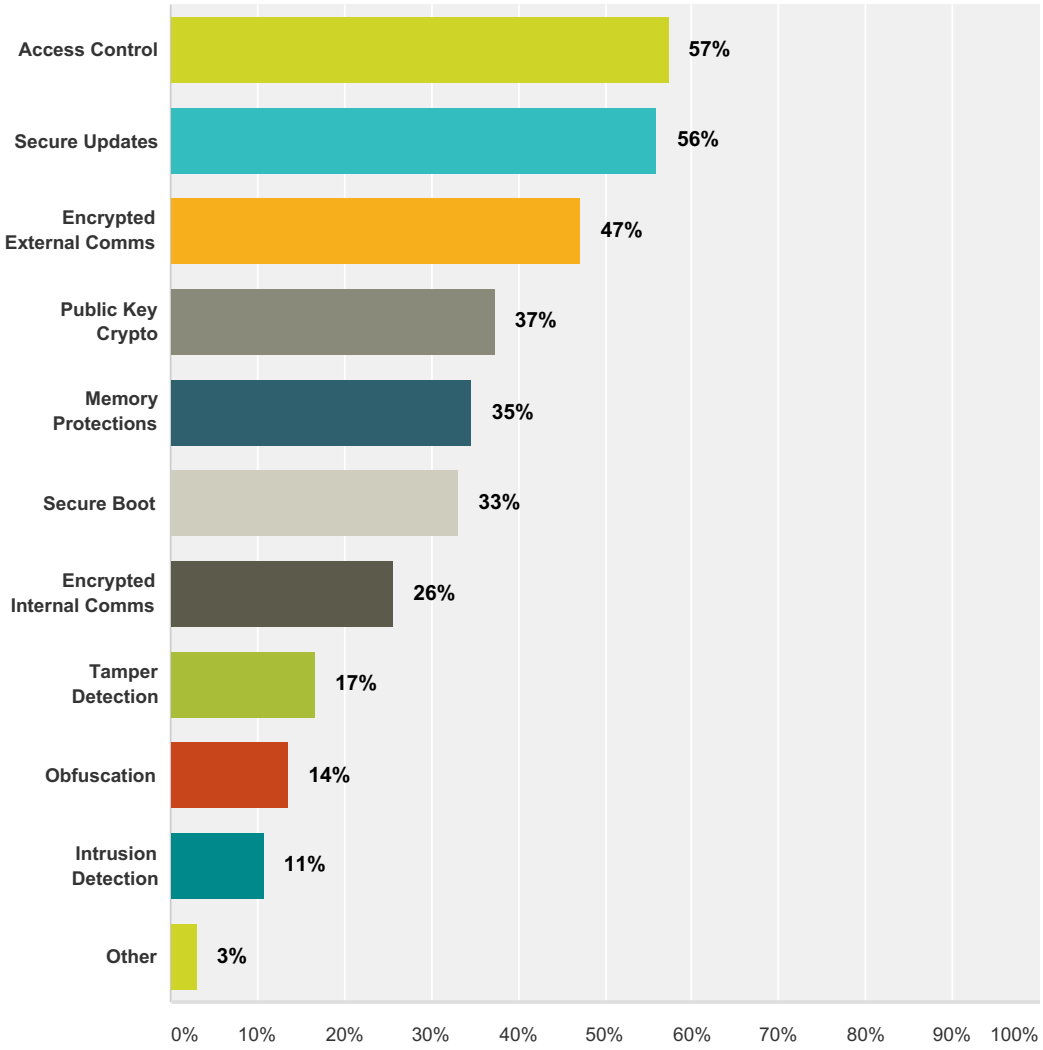
Answer Choices	Responses	
Product Tampering	57%	558
Theft of Data	41%	399
Theft of IP	37%	365
Privacy Violations	36%	350
Product Cloning	33%	329
Denial of Service	32%	314
Injury or Death	26%	258
Theft of Service	18%	179
Blackmail or Ransom	5%	46
Other	2%	23

Total Respondents: 983	

#	Other (please specify)	Date
1	large scale equipment damage, improper operation	1/25/2017 11:50 PM
2	Failed quality control of a safety related procedure.q	1/25/2017 11:25 PM
3	Property theft	1/25/2017 11:18 PM
4	x	1/25/2017 8:33 PM
5	Changing setting to invalidate calibration	1/25/2017 4:45 PM
6	access by unauthorized persons	1/20/2017 9:09 AM
7	confidential	1/18/2017 8:37 PM
8	Incorrect operation	1/18/2017 6:08 PM
9	Need to bring others on board to consider threats that haven't been considered.	1/18/2017 5:47 PM
10	Cannot disclose	1/18/2017 4:51 PM
11	MiM	1/17/2017 9:33 AM
12	Illicit SIM use	1/17/2017 8:39 AM
13	Hipaa specifically; adverse event reporting, analysis, etc.	1/16/2017 10:12 PM
14	Decline to answer at present time. Happy to discuss, but responding to this question is a security risk!	1/16/2017 6:59 PM
15	keep product reliable and working	1/14/2017 3:06 AM
16	Product reliability is the real security concern ie fail safe	1/13/2017 7:15 AM
17	reliability	1/13/2017 6:46 AM
18	Intrusion	1/12/2017 10:30 PM
19	I can't talk about it	1/12/2017 7:26 PM
20	Entry point for atackers!	1/12/2017 5:01 PM
21	Correct implementation of Apple's HAP security protocol	1/12/2017 4:43 PM
22	Data Tampering for nefarious purposes	1/11/2017 10:42 PM
23	I don't know	1/11/2017 2:43 PM

Q24 If you know, which of the following security layers are used on your current project? (select all that apply)

Answered: 935 Skipped: 791



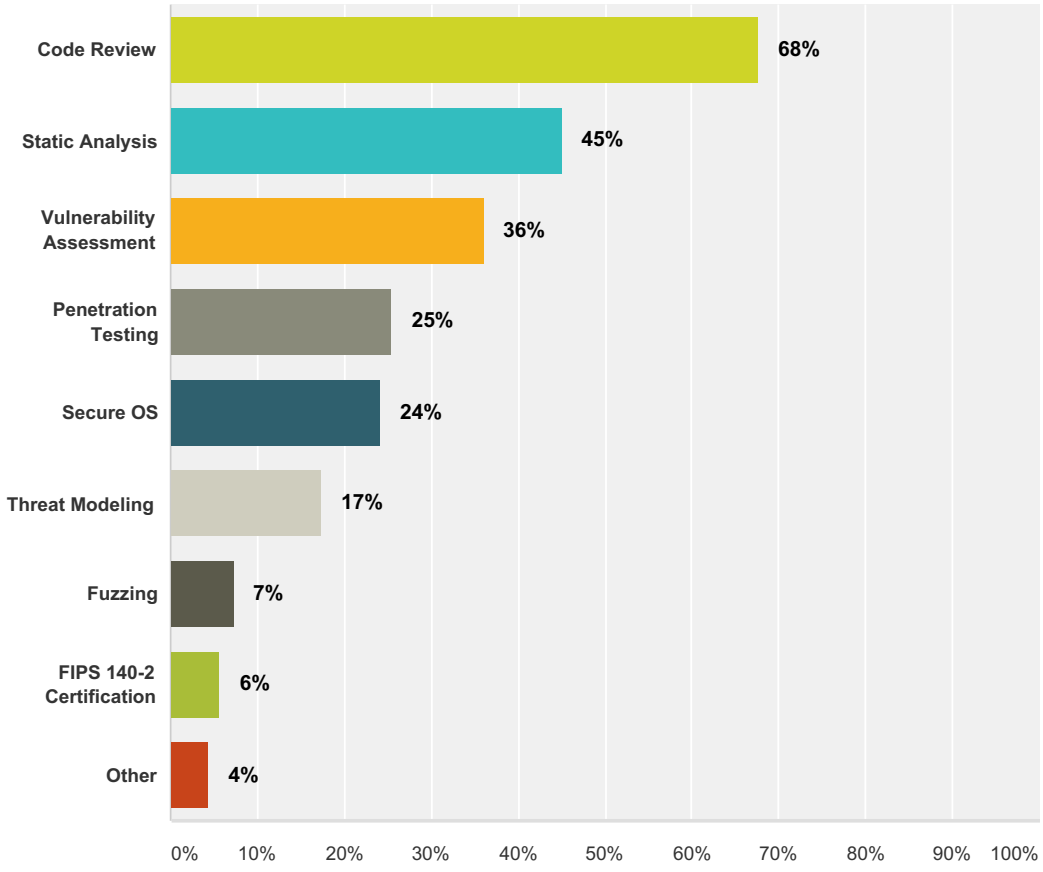
Answer Choices	Responses	
Access Control	57%	536
Secure Updates	56%	523
Encrypted External Comms	47%	442
Public Key Crypto	37%	349
Memory Protections	35%	325
Secure Boot	33%	311
Encrypted Internal Comms	26%	240

Tamper Detection	**17%**	156
Obfuscation	**14%**	127
Intrusion Detection	**11%**	102
Other	**3%**	29
Total Respondents: 935		

#	Other (please specify)	Date
1	Authenticated Internal Communications	1/31/2017 11:49 AM
2	Not disclosing	1/26/2017 6:06 PM
3	Deployed binary, HW, SW key	1/26/2017 5:59 PM
4	significant physical security, minimalized interface	1/25/2017 11:50 PM
5	One way hashing	1/25/2017 11:18 PM
6	x	1/25/2017 8:33 PM
7	Absence of ethernet connection	1/25/2017 4:33 PM
8	self recovery on failure	1/20/2017 7:42 PM
9	Use of VPNs over cellular	1/19/2017 7:21 PM
10	don't know yet	1/19/2017 7:17 PM
11	confidential	1/18/2017 8:37 PM
12	Embedded Model Comparison of Expected Behavior	1/18/2017 6:19 PM
13	Custom	1/18/2017 6:08 PM
14	follow ISA/IEC-62443/ISA-99 - looking into auditing	1/18/2017 5:40 PM
15	Don't discuss security implementations	1/18/2017 5:34 PM
16	Cannot disclose	1/18/2017 4:51 PM
17	Not sure, I have no background on cryptography	1/17/2017 10:10 PM
18	Physical inaccessibility	1/17/2017 9:45 AM
19	Can not answer	1/14/2017 10:23 PM
20	Whatever is built into the NXP Zigbee stack	1/13/2017 2:00 AM
21	I can't talk about it	1/12/2017 7:26 PM
22	any and all	1/12/2017 6:24 PM
23	Additional Client measures not part of this project.	1/12/2017 5:54 PM
24	Firewalling	1/12/2017 5:20 PM
25	Study to use virtualization and containerization ongoing	1/12/2017 5:01 PM
26	password protection of GUI	1/12/2017 4:18 PM
27	No O/S, No remote firmware updating, No ability to remotely access firmware or memory. Remote connectivity limited to text menu user I/F.	1/12/2017 4:15 PM
28	Not at liberty to say	1/11/2017 6:13 PM
29	shutting down ethernet during normal device operation.	1/10/2017 7:55 PM

Q25 If you know, which of the following processes are used to increase security on your current project? (select all that apply)

Answered: 802 Skipped: 924



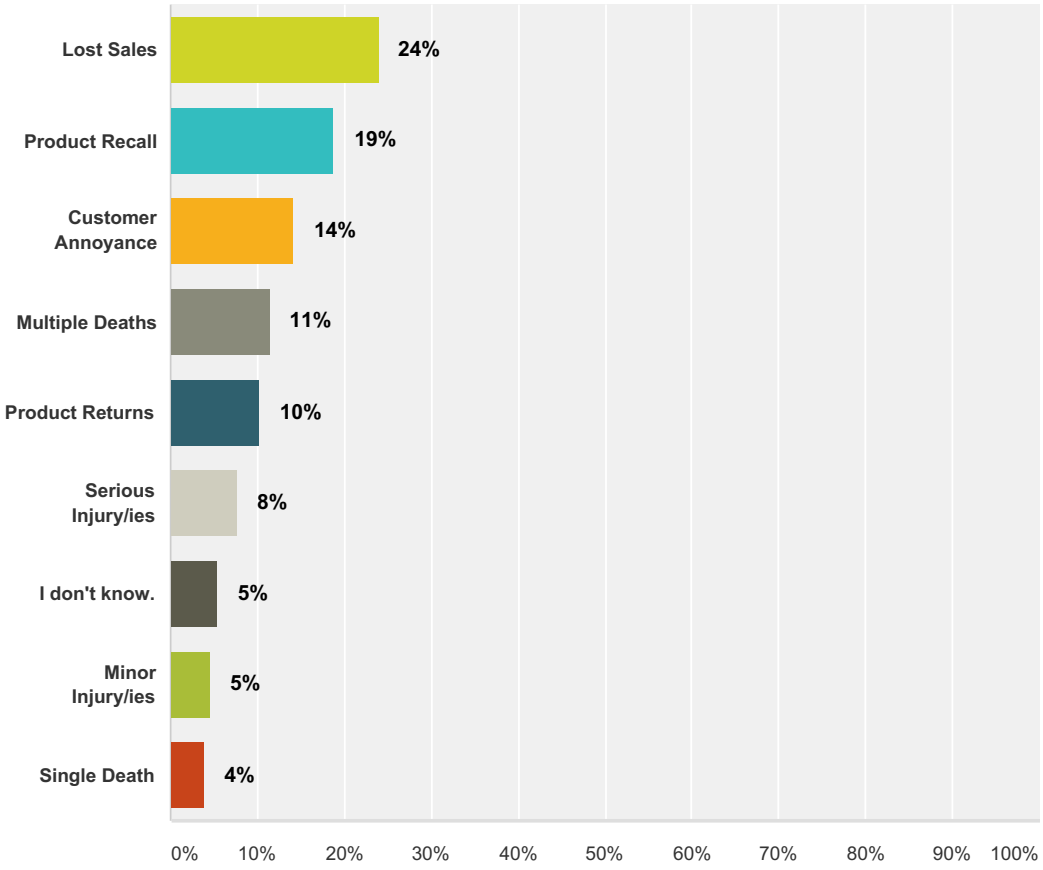
Answer Choices	Responses	
Code Review	68%	542
Static Analysis	45%	361
Vulnerability Assessment	36%	289
Penetration Testing	25%	204
Secure OS	24%	194
Threat Modeling	17%	139
Fuzzing	7%	58
FIPS 140-2 Certification	6%	46
Other	4%	35
Total Respondents: 802		

#	Other (please specify)	Date

1	None	1/29/2017 5:20 PM
2	undecided	1/27/2017 7:43 PM
3	Complete isolation from public Internet	1/27/2017 5:34 AM
4	Not disclosing	1/26/2017 6:06 PM
5	x	1/25/2017 8:33 PM
6	only connect to internet for support	1/25/2017 4:56 PM
7	SHE	1/25/2017 1:15 PM
8	The master to which the current system is connected to, takes care of it.	1/20/2017 5:19 AM
9	Third Party Tester	1/19/2017 8:01 PM
10	don't know yet	1/19/2017 7:17 PM
11	at present customised techniques	1/19/2017 2:05 PM
12	TPM	1/19/2017 5:09 AM
13	Security key lock for module protection	1/18/2017 10:58 PM
14	confidential	1/18/2017 8:37 PM
15	ECDF	1/18/2017 6:13 PM
16	Applying industry standards	1/18/2017 6:08 PM
17	Additional	1/18/2017 5:47 PM
18	ISA/IEC-62443/ISA-99	1/18/2017 5:40 PM
19	Don't discuss security implementations	1/18/2017 5:34 PM
20	Tpm	1/18/2017 5:08 PM
21	Not sure, I have no background on cryptography	1/17/2017 10:10 PM
22	OEM will test (they have been hurt by CERT reports)	1/17/2017 5:45 PM
23	Don't know	1/17/2017 12:07 PM
24	May be others but I am not involved	1/17/2017 2:00 AM
25	Design to limit vulnerabilities	1/16/2017 7:02 PM
26	Can not answer	1/14/2017 10:23 PM
27	Don't know - that part is not what I'm working on	1/13/2017 11:39 AM
28	Vendor supplied	1/13/2017 2:00 AM
29	protocol specification implementation	1/12/2017 10:28 PM
30	I can't talk about it	1/12/2017 7:26 PM
31	Customer-reported issues	1/12/2017 7:20 PM
32	Reliance on SSL vendor	1/12/2017 4:40 PM
33	password protection of GUI	1/12/2017 4:18 PM
34	common sense	1/12/2017 4:15 PM
35	Not at liberty to say	1/11/2017 6:13 PM

Q26 If the product resulting from your current project malfunctioned, what is the worst possible outcome?

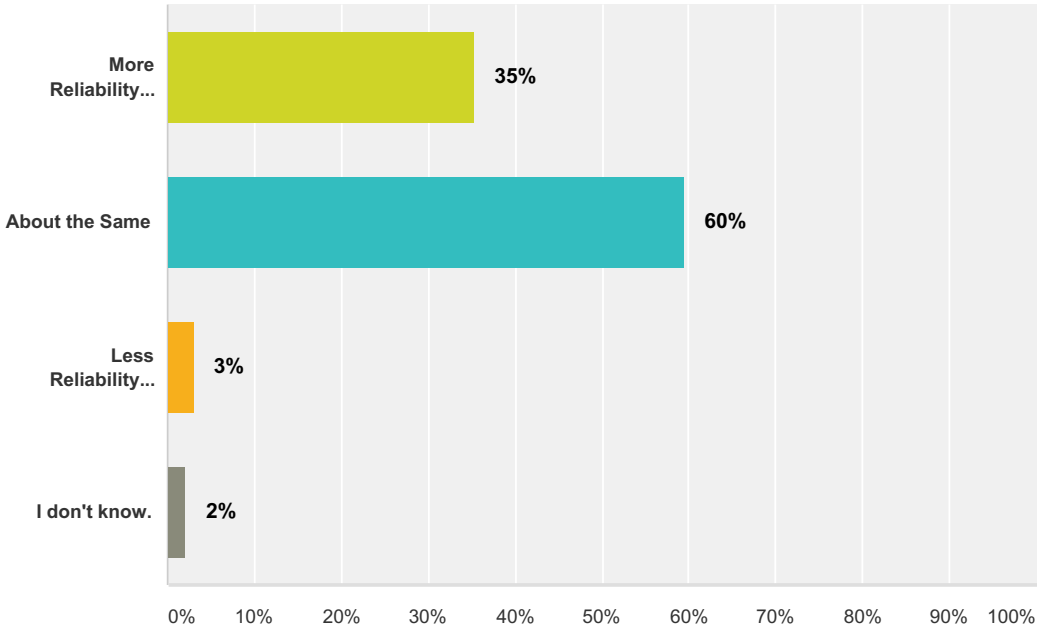
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
Lost Sales	24%	413
Product Recall	19%	324
Customer Annoyance	14%	246
Multiple Deaths	11%	197
Product Returns	10%	175
Serious Injury/ies	8%	132
I don't know.	5%	93
Minor Injury/ies	5%	78
Single Death	4%	68
Total		**1,726**

Q27 How do the reliability needs of your current project compare to your other recent projects?

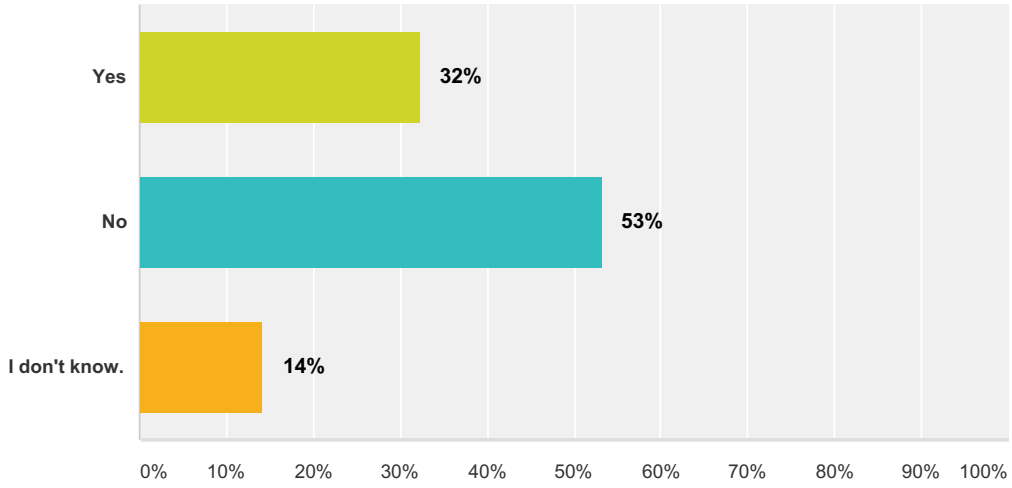
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
More Reliability Required	**35%**	608
About the Same	**60%**	1,027
Less Reliability Required	**3%**	54
I don't know.	**2%**	37
Total		**1,726**

Q28 Will your current project be designed in accordance with a safety standard (e.g., FDA, DO-178, IEC-61508)?

Answered: 1,726 Skipped: 0



Answer Choices	Responses	
Yes	**32%**	560
No	**53%**	920
I don't know.	**14%**	246
Total		**1,726**

#	Name(s) of applicable standards, if any:	Date
1	ISO26262	2/2/2017 7:18 AM
2	IEC-61508	1/31/2017 6:57 PM
3	DO-178	1/31/2017 6:07 PM
4	61508	1/31/2017 6:06 PM
5	26262	1/31/2017 4:16 PM
6	ISO-26262	1/31/2017 3:40 AM
7	IEC-61508	1/30/2017 10:28 PM
8	iec 61010-1	1/30/2017 9:20 PM
9	C tick	1/30/2017 2:04 PM
10	IEC-61508	1/27/2017 8:22 PM
11	CENELEC 50126, 50128, 50129, AREMA	1/26/2017 6:01 PM
12	ISO 26262	1/26/2017 9:12 AM
13	FDA / Class 2 Medical Device	1/25/2017 9:30 PM
14	Product part of ceiling light fixture 0-10V out controls lamp output	1/25/2017 7:44 PM
15	FDA, EU medical device standards, 62304	1/25/2017 5:48 PM
16	IEC-61508	1/25/2017 5:19 PM

17	FDA	1/25/2017 4:57 PM
18	UL325	1/25/2017 4:36 PM
19	ISO26262	1/25/2017 1:17 PM
20	IEC-60601, FDA 510k	1/23/2017 6:41 PM
21	similar to ISO26262	1/23/2017 9:08 AM
22	IEC-60601	1/20/2017 8:29 PM
23	IEC 60601-1, 13485	1/19/2017 7:50 PM
24	FDA, various IECs	1/19/2017 7:21 PM
25	ISO26262	1/19/2017 6:47 PM
26	UL 3rd part testing	1/19/2017 6:27 PM
27	IEC-61508, ATEX, and various FM and UL requirements	1/19/2017 4:43 PM
28	ISO 13485, ISO 14971, IEC 62304	1/19/2017 3:52 PM
29	IEC-61508	1/19/2017 2:21 PM
30	ISO 26262	1/19/2017 2:16 PM
31	IEC-61508, ISO-13849...	1/19/2017 10:26 AM
32	DO-254, ARINC653	1/19/2017 8:39 AM
33	ISO26262	1/19/2017 7:25 AM
34	ISO 26262	1/19/2017 4:13 AM
35	IEC-61508	1/19/2017 1:33 AM
36	Iso26262	1/18/2017 11:02 PM
37	confidential	1/18/2017 8:38 PM
38	CSA class 1 Div C, D	1/18/2017 8:02 PM
39	DO-178 C	1/18/2017 7:07 PM
40	60601	1/18/2017 6:46 PM
41	That's another department....	1/18/2017 6:10 PM
42	ISO26262	1/18/2017 5:26 PM
43	ASME A17.1	1/18/2017 5:25 PM
44	ISO 26262	1/18/2017 5:25 PM
45	IEC-60335	1/18/2017 3:45 PM
46	iso 26262	1/18/2017 8:36 AM
47	UL858,60730	1/17/2017 11:45 PM
48	IEC 62443-2-4, NIST SP800-53	1/17/2017 5:48 PM
49	UL95	1/17/2017 5:41 PM
50	NASA-STD-8719.13 Software Safety Standard	1/17/2017 5:29 PM
51	Several safety standards	1/17/2017 12:09 PM
52	ISO-26262	1/17/2017 10:24 AM
53	IEC-61508	1/17/2017 10:17 AM
54	DO-178	1/17/2017 9:47 AM
55	UL 61010-1	1/17/2017 9:39 AM
56	FDA	1/16/2017 10:49 PM
57	iec-61010	1/16/2017 10:14 PM

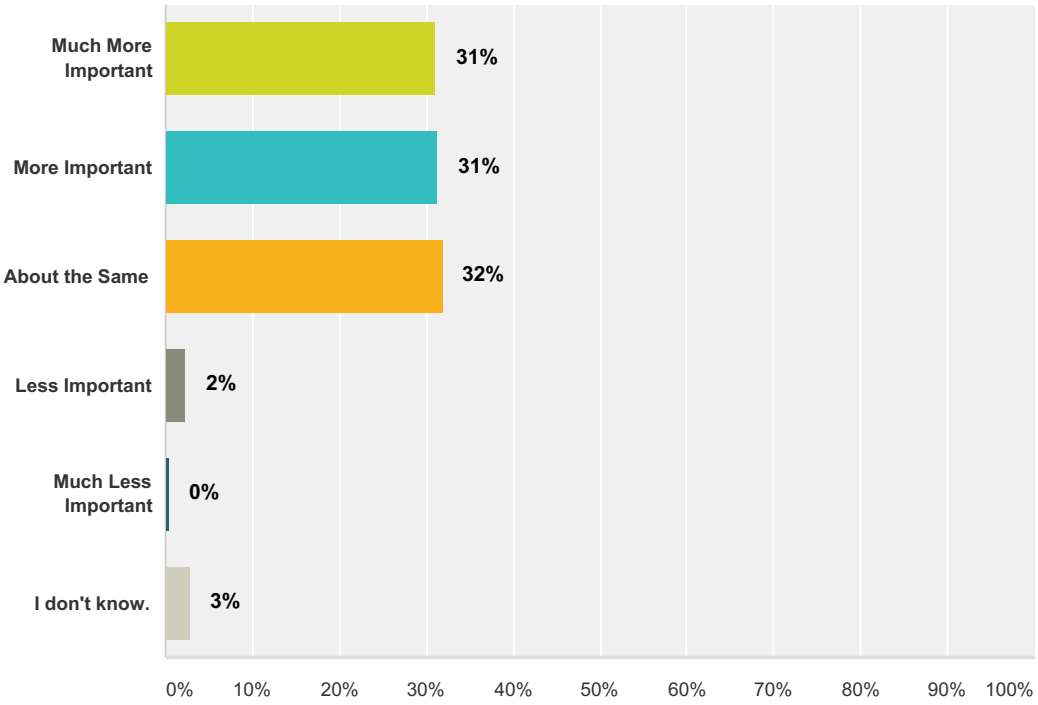
58	FDA	1/16/2017 10:13 PM
59	UL	1/16/2017 9:34 PM
60	ISO 26262	1/16/2017 9:31 PM
61	IEC 60601-1	1/16/2017 8:49 PM
62	do-178	1/16/2017 7:51 PM
63	iso26262	1/16/2017 7:48 PM
64	60601	1/16/2017 7:03 PM
65	The product will seek IEC-61508 compliance in the future.	1/16/2017 7:01 PM
66	FDA	1/16/2017 6:37 PM
67	IEC 60079-29-1 EN50271	1/16/2017 6:33 PM
68	IEC-60335	1/16/2017 5:56 PM
69	NF EN 50128	1/16/2017 5:27 PM
70	ISO26262	1/16/2017 12:40 PM
71	IEC 61508	1/16/2017 11:06 AM
72	DO178B Level A	1/16/2017 2:57 AM
73	EN-15194	1/15/2017 7:48 PM
74	IEC-61508	1/15/2017 1:43 PM
75	ISO-26262	1/15/2017 11:51 AM
76	Fda	1/14/2017 10:15 AM
77	Iso26262	1/14/2017 9:44 AM
78	DO-178B	1/14/2017 2:40 AM
79	UL 1069	1/14/2017 2:00 AM
80	ISO 26262	1/13/2017 7:42 PM
81	IEC-60601, IEC-62304	1/13/2017 5:26 PM
82	IEC-60601	1/13/2017 4:30 PM
83	ISO26262	1/13/2017 1:01 PM
84	ISO 60601 and related standards	1/13/2017 11:40 AM
85	IEC-60601	1/13/2017 8:33 AM
86	DO-178 IEC-61508	1/13/2017 7:16 AM
87	ISO 26262	1/13/2017 6:31 AM
88	ISO26262	1/13/2017 6:18 AM
89	62304	1/13/2017 5:58 AM
90	UL standards (not sure which one)	1/13/2017 4:10 AM
91	ISO26262	1/13/2017 4:09 AM
92	ISO 26262	1/13/2017 2:27 AM
93	DO-178	1/13/2017 12:10 AM
94	IEC-61508	1/13/2017 12:05 AM
95	IEC-61508	1/12/2017 11:20 PM
96	DO-178, DO-260	1/12/2017 10:29 PM
97	IEC standards and UL standards for the product	1/12/2017 10:22 PM
98	UL 61010	1/12/2017 10:03 PM

99	IEC60335	1/12/2017 9:06 PM
100	ISO26262	1/12/2017 8:54 PM
101	DNV	1/12/2017 8:28 PM
102	62304	1/12/2017 8:15 PM
103	IEC-61508: SIL 2 level	1/12/2017 8:14 PM
104	ISO 13849	1/12/2017 8:14 PM
105	FDA	1/12/2017 6:43 PM
106	EN62304	1/12/2017 6:35 PM
107	FDA	1/12/2017 6:18 PM
108	IEC-61508 SIL2	1/12/2017 5:55 PM
109	60601-1,62304	1/12/2017 5:34 PM
110	EN-16590	1/12/2017 5:31 PM
111	ISO26262	1/12/2017 5:25 PM
112	hazardous locations cULus, ATEX, IECEx	1/12/2017 5:12 PM
113	ISO 26262	1/12/2017 5:01 PM
114	none relate to software. we enforce safety with hardware	1/12/2017 4:53 PM
115	EN-50128:2011	1/12/2017 4:47 PM
116	Automotive SPICE	1/12/2017 4:40 PM
117	Iec 60730	1/12/2017 4:40 PM
118	62304	1/12/2017 4:40 PM
119	Company established	1/12/2017 4:38 PM
120	61508	1/12/2017 4:37 PM
121	ISO 26262	1/12/2017 4:27 PM
122	ISO 26262	1/12/2017 4:22 PM
123	IEC-61508	1/12/2017 4:19 PM
124	applicable military standards	1/12/2017 4:16 PM
125	IEC-61508	1/12/2017 4:15 PM
126	UL60950, UL962, FCC47	1/12/2017 4:15 PM
127	UL	1/12/2017 4:12 PM
128	UL 2595	1/12/2017 4:11 PM
129	FDA IEC 62304	1/12/2017 4:11 PM
130	UL - Class B	1/12/2017 2:24 PM
131	IEC-61508, EN-6024-1	1/12/2017 12:59 PM
132	ISO 26262	1/11/2017 10:00 PM
133	62304, FDA	1/11/2017 6:28 PM
134	60601, FDA guidence	1/11/2017 1:13 PM
135	iso26262 asil a	1/11/2017 11:07 AM
136	ISO26262	1/11/2017 11:07 AM
137	EN50128	1/11/2017 10:55 AM
138	EN-50126, EN-50128, EN-50129, EN-50159	1/11/2017 9:44 AM
139	IEC61508	1/11/2017 9:28 AM

140	iso 26262	1/10/2017 8:56 PM
141	IEC-61508	1/10/2017 7:52 PM
142	No but will use development process like those called out in DO-178	1/10/2017 7:46 PM
143	IEC 62304 - FDA Guidance for Premarket Notification	1/10/2017 7:44 PM
144	ISO26262	1/10/2017 7:43 PM
145	IEC 61511	1/10/2017 7:40 PM
146	Eventually we will probably look at getting a IEC 61508 SIL 3 rating as well as	1/10/2017 7:32 PM
147	MIL-882 and IEEE standards	1/10/2017 7:05 PM
148	iso-26262	1/10/2017 7:01 PM

Q29 What is the relative importance of reliability vs. security on your current project?

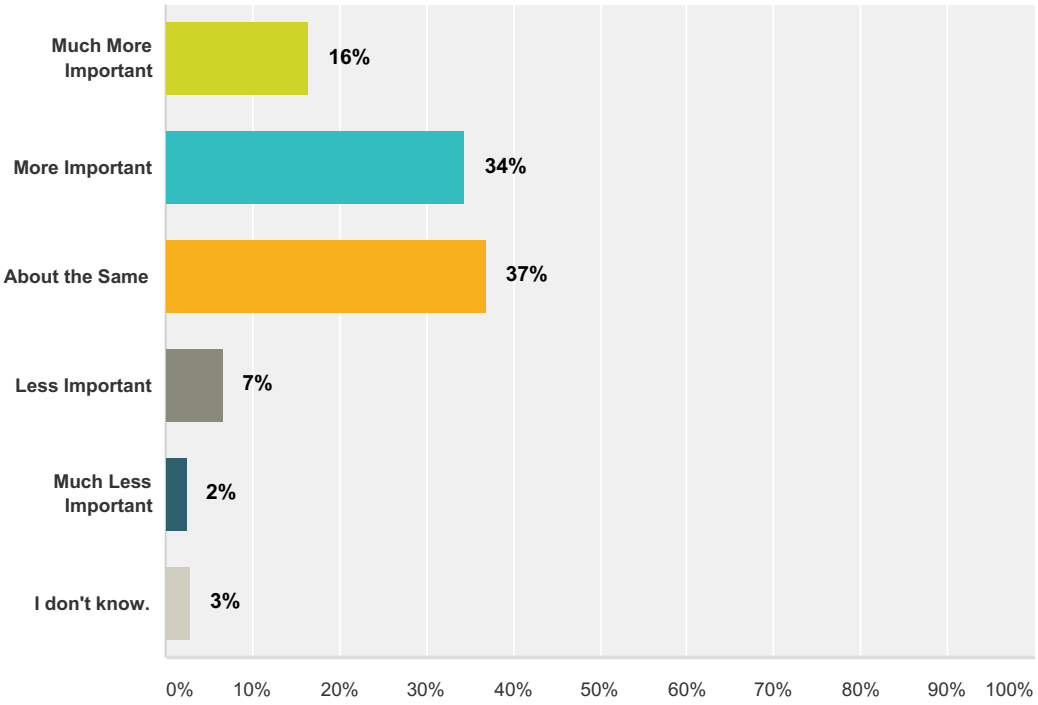
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
Much More Important	31%	537
More Important	31%	539
About the Same	32%	553
Less Important	2%	40
Much Less Important	0%	6
I don't know.	3%	51
Total		1,726

Q30 What is the relative importance of reliability vs. meeting the schedule on your current project?

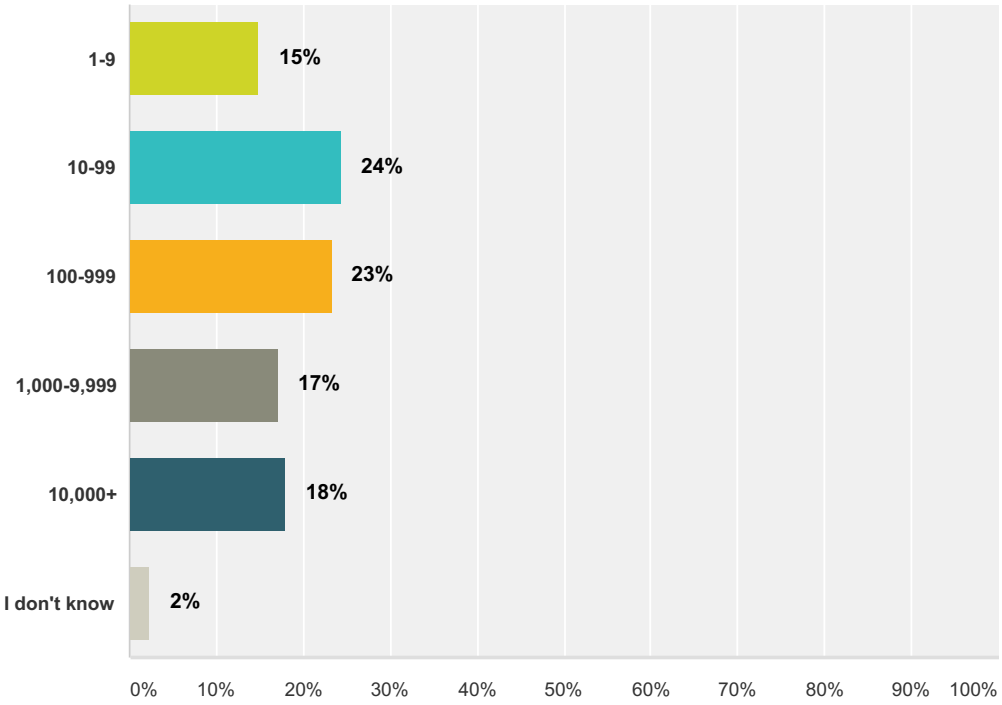
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
Much More Important	16%	284
More Important	34%	593
About the Same	37%	639
Less Important	7%	117
Much Less Important	2%	42
I don't know.	3%	51
Total		1,726

Q31 Approximately how many total people work at your company (across all locations)?

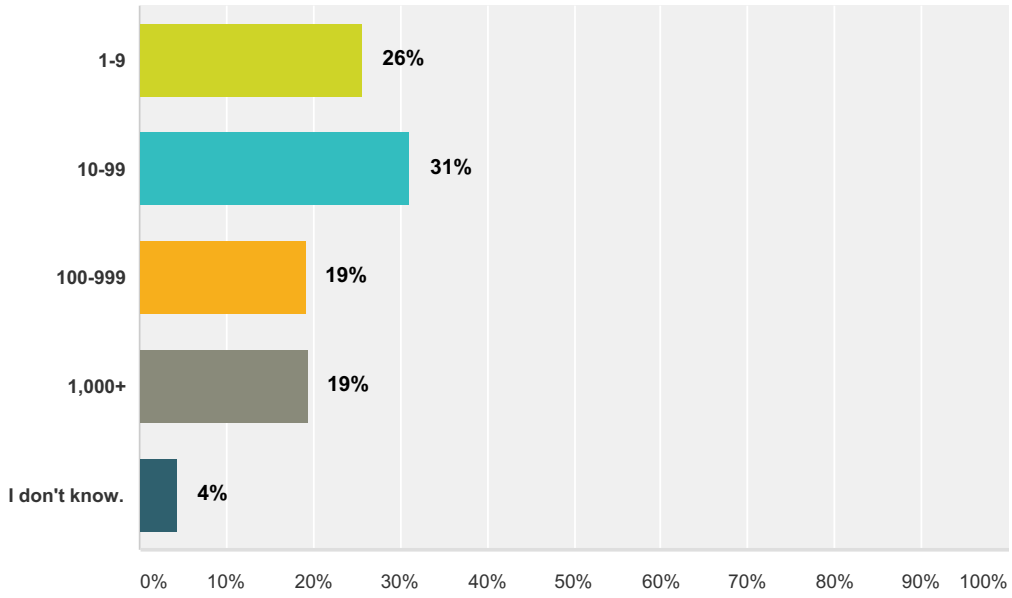
Answered: 1,726 Skipped: 0



Answer Choices	Responses	
1-9	**15%**	256
10-99	**24%**	422
100-999	**23%**	405
1,000-9,999	**17%**	295
10,000+	**18%**	309
I don't know	**2%**	39
Total		**1,726**

Q32 Approximately how many engineers (of any type) work at the company?

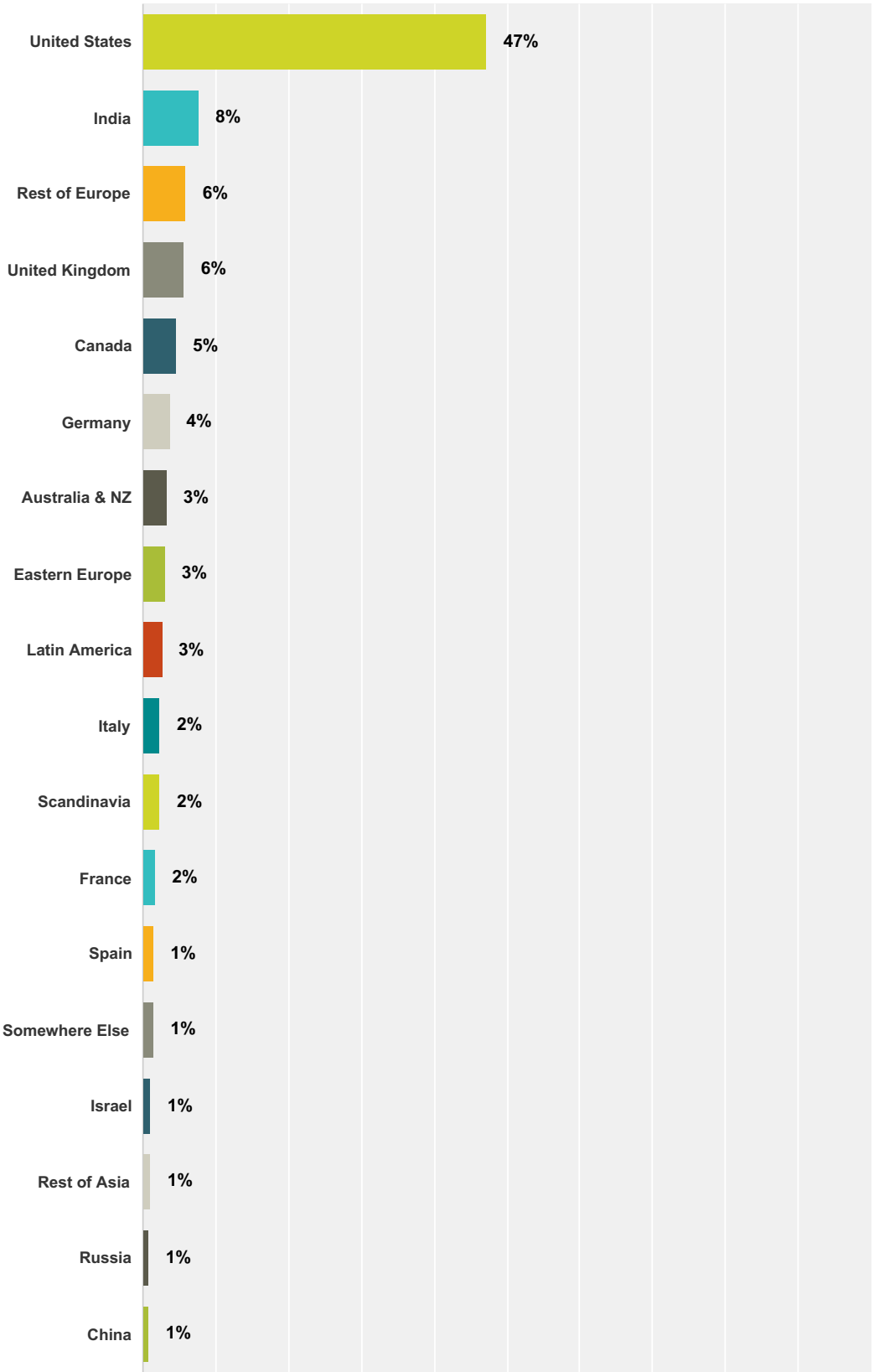
Answered: 1,726 Skipped: 0

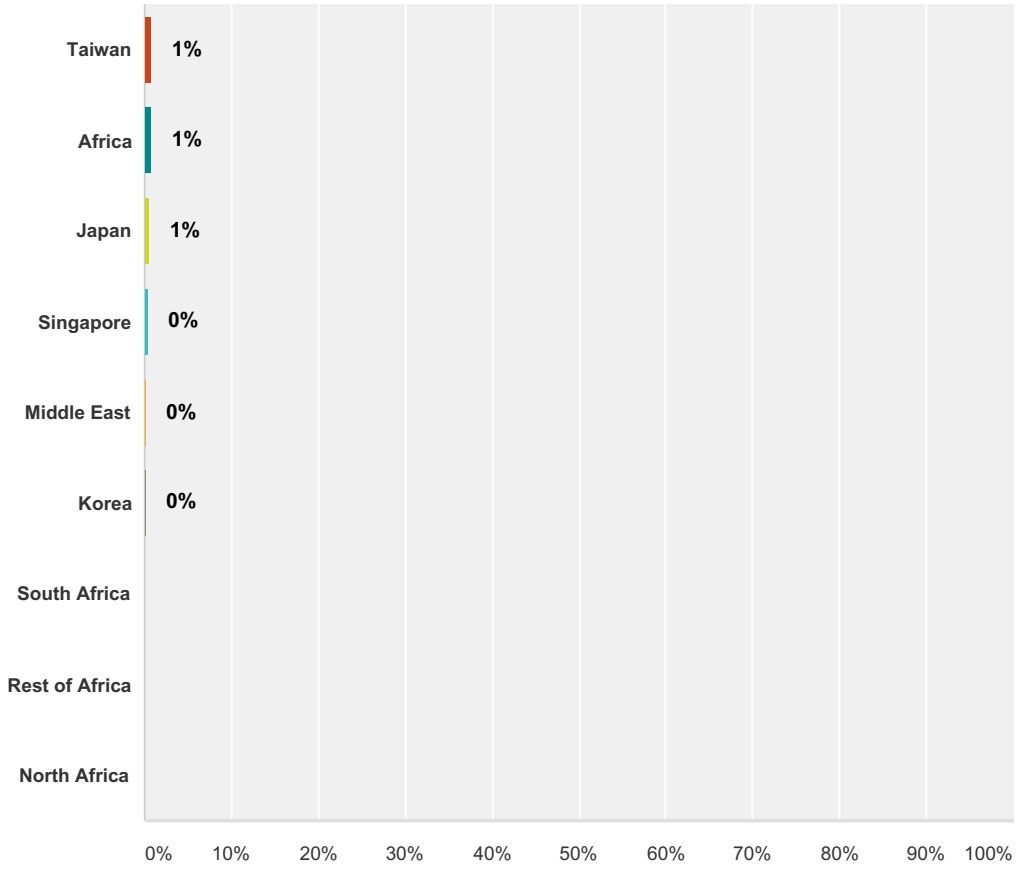


Answer Choices	Responses	
1-9	**26%**	444
10-99	**31%**	536
100-999	**19%**	333
1,000+	**19%**	336
I don't know.	**4%**	77
Total		**1,726**

Q33 Which of the following countries or regions best describes where you currently reside?

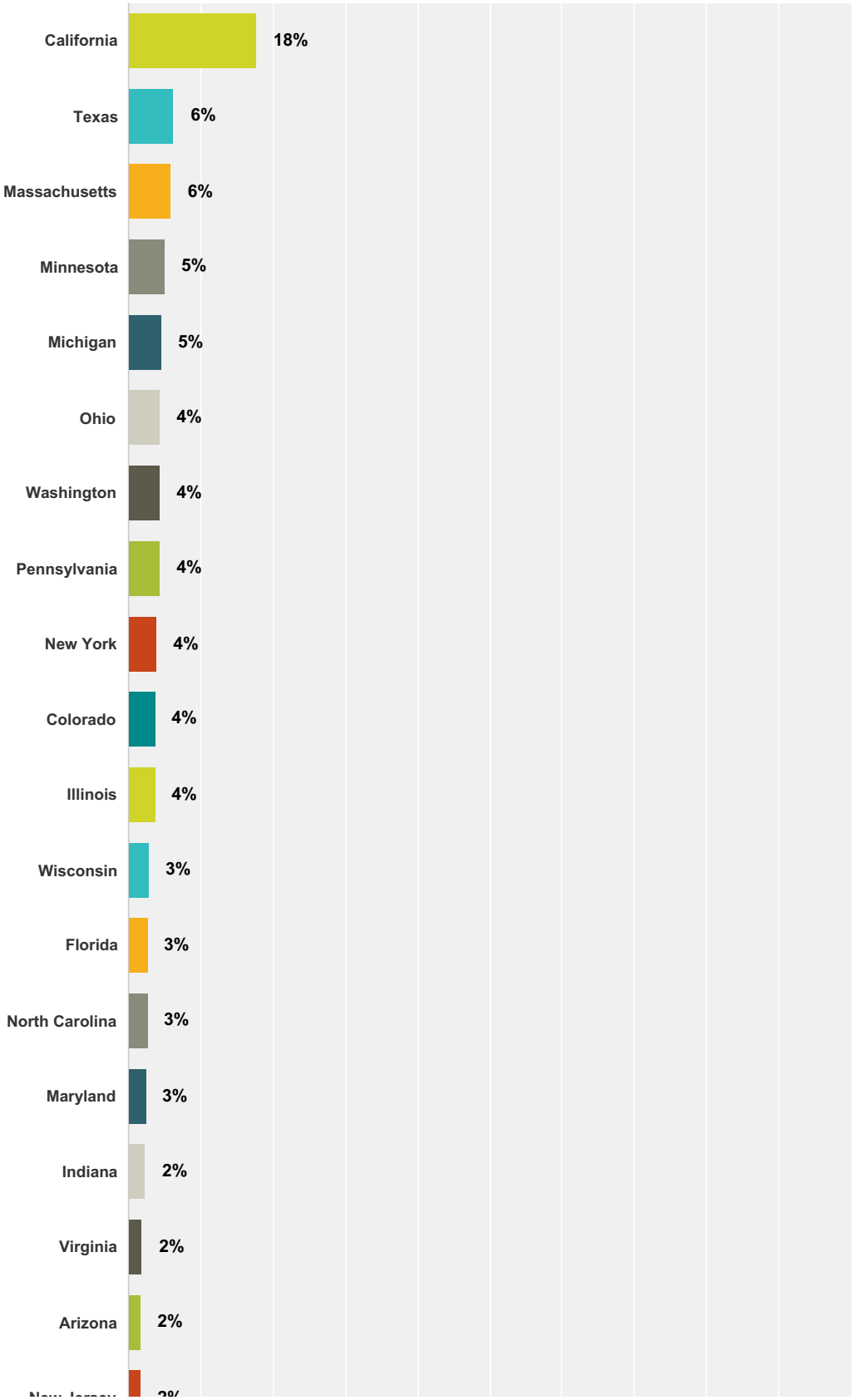
Answered: 1,723 Skipped: 3

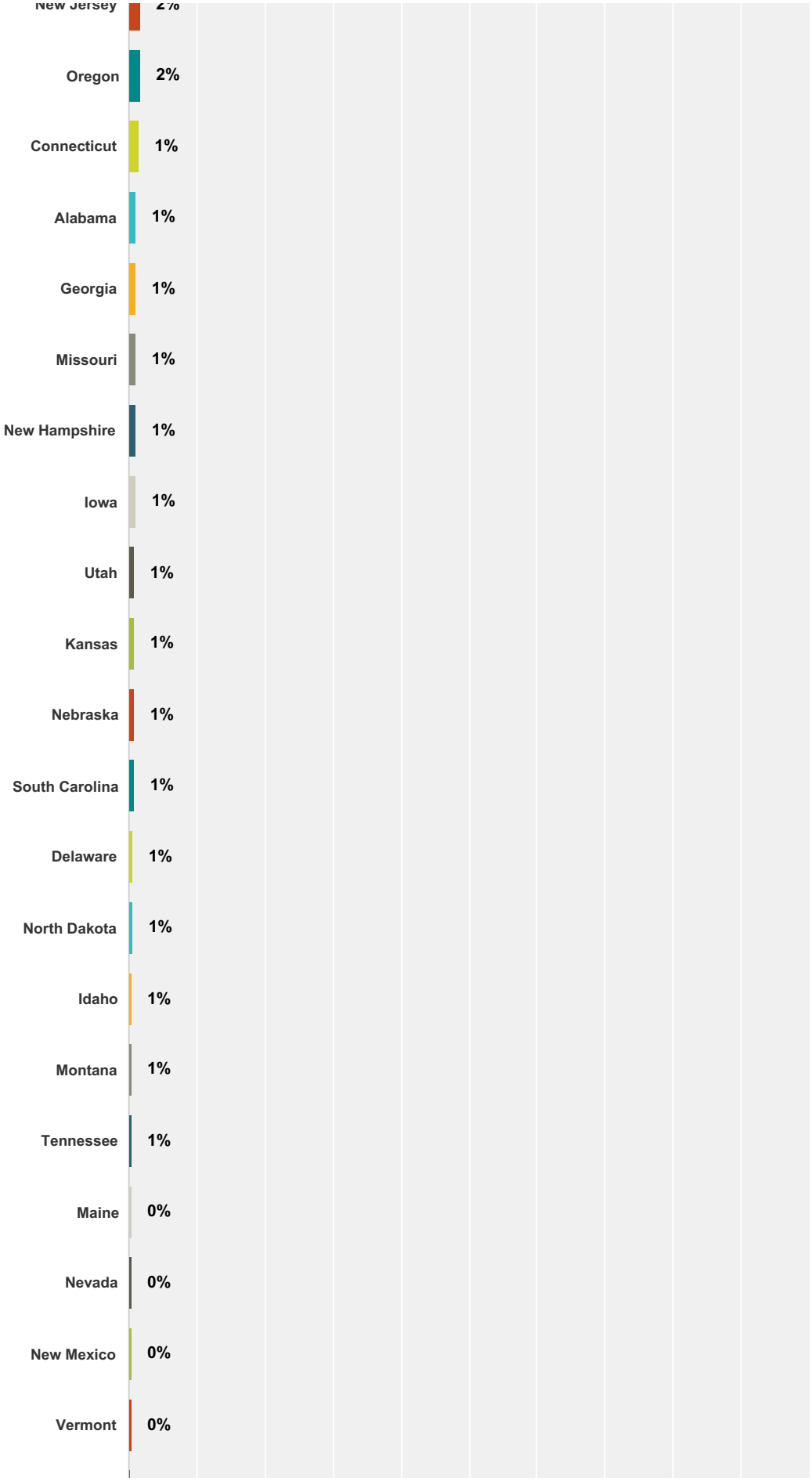


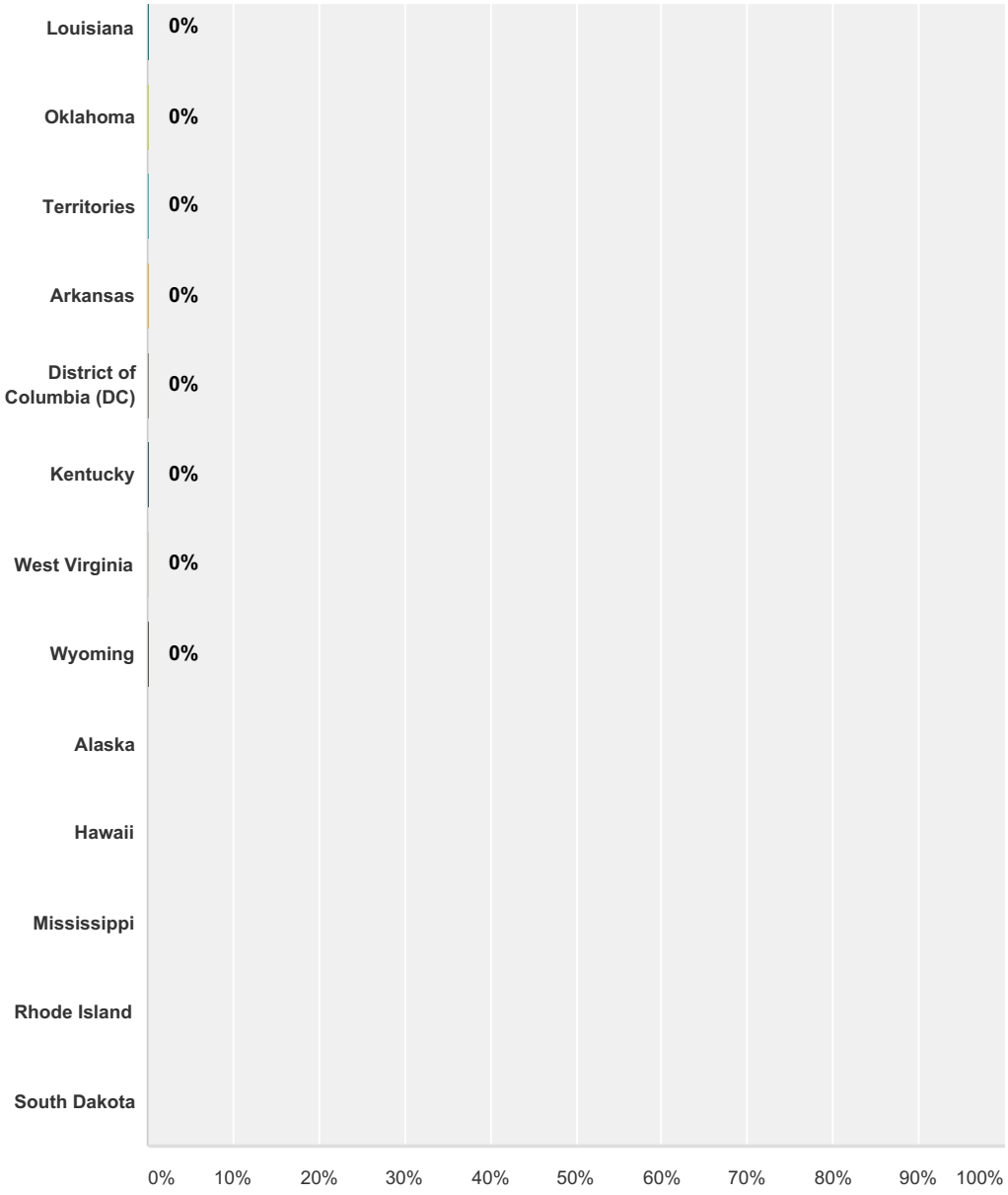

Taiwan
1%
Africa
1%
Japan
1%
Singapore
0%
Middle East
0%
Korea
0%
South Africa
Rest of Africa
North Africa
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%

Q34 In which part of the United States do you currently reside?

Answered: 793 Skipped: 933




New Jersey 2%
Oregon 2%
Connecticut 1%
Alabama 1%
Georgia 1%
Missouri 1%
New Hampshire 1%
Iowa 1%
Utah 1%
Kansas 1%
Nebraska 1%
South Carolina 1%
Delaware 1%
North Dakota 1%
Idaho 1%
Montana 1%
Tennessee 1%
Maine 0%
Nevada 0%
New Mexico 0%
Vermont 0%


Louisiana
0%
Oklahoma
0%
Territories
0%
Arkansas
0%
District of Columbia (DC)
0%
Kentucky
0%
West Virginia
0%
Wyoming
0%
Alaska
Hawaii
Mississippi
Rhode Island
South Dakota
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%

Barr Group

2017

Embedded Systems Safety & Security Survey

Data Licensing Contact:

Andrew Girson, CEO
agirson@barrgroup.com
+1 (866) 65-EMBED