

Embedded Systems Survey 2025: Insights and Trends

PREPARED BY:

Maurizio Di Paolo Emilio, Ph.D

Editor

Embedded.com

embedded
survey ✓

Introduction

- ✓ **Global survey of embedded engineers** – Captures insights from professionals worldwide.
- ✓ **Focus on roles, projects, tools, and technologies** – Provides a holistic view of embedded development.
- ✓ **Foundation for industry insights and trends** – Establishes key data for understanding current and future directions.



Methodology

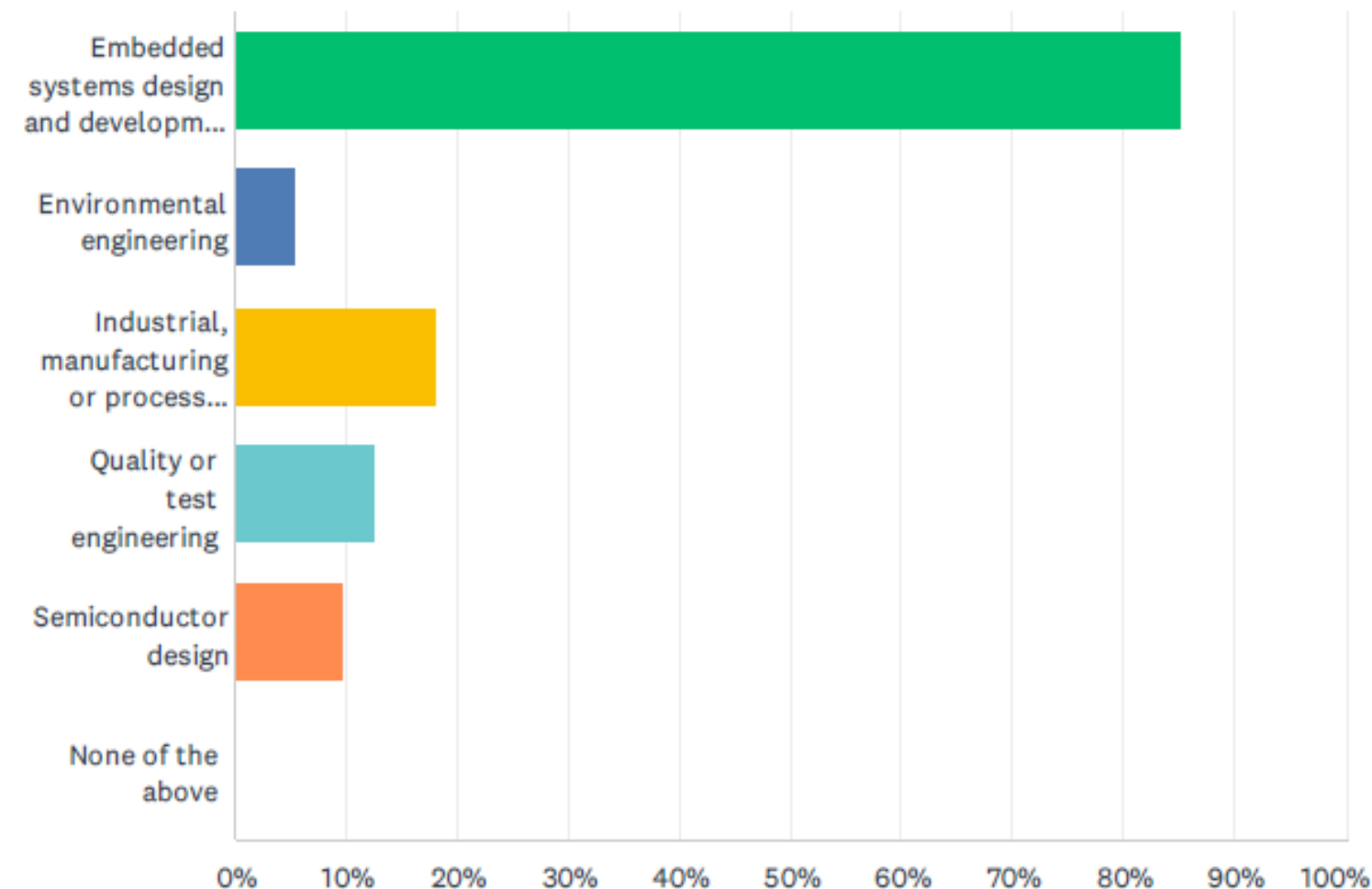
Online Survey

- **Field Dates:** June 9 to August 31, 2025
- Respondents screened for **engineering** responsibilities and **experience with embedded applications**
- Audience supplied by **ASPENCORE**
- Results based on **424 responses**
 - **EMEA 46%**
 - **Americas 40%**
 - **APAC 14%**

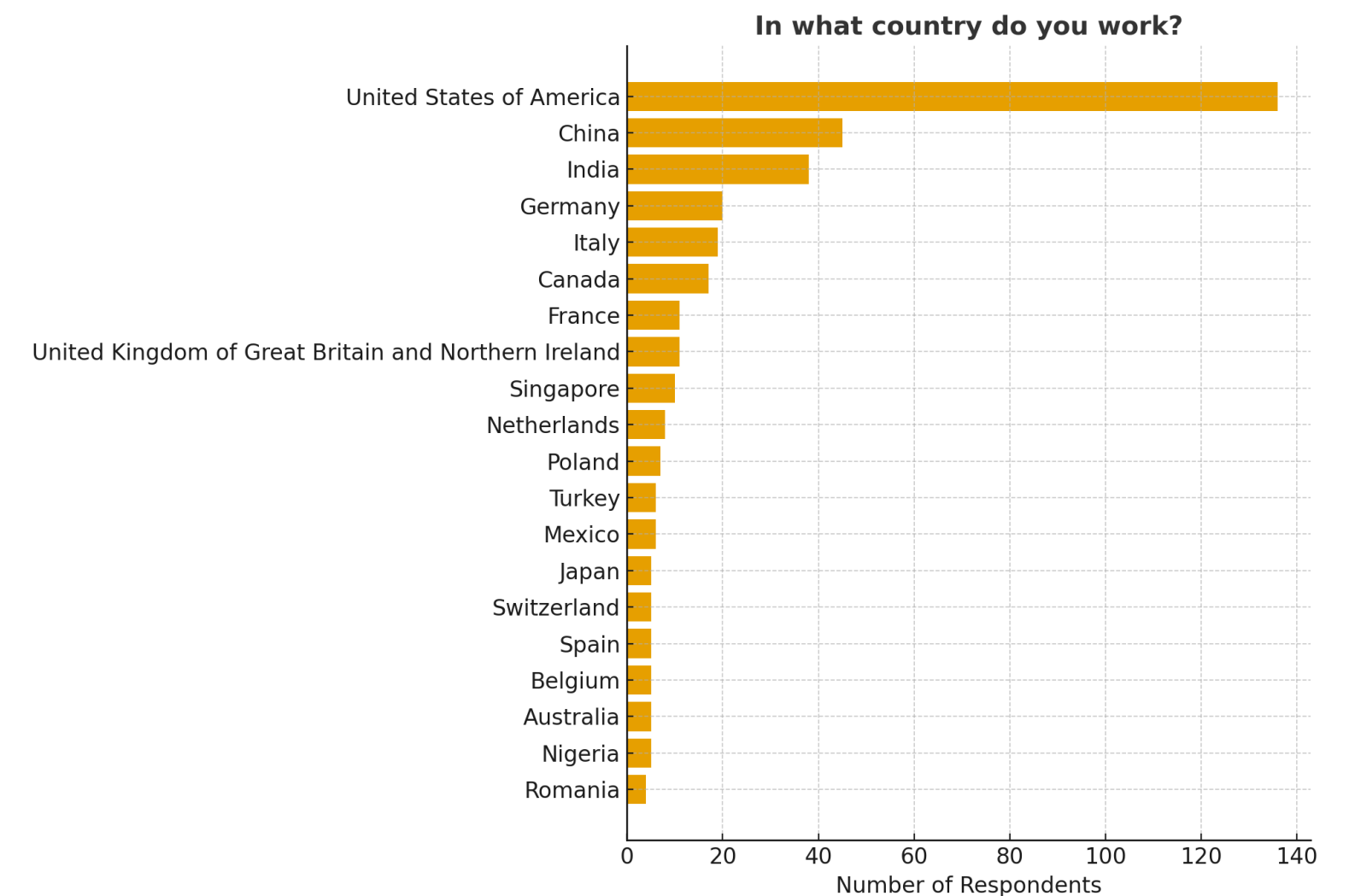
Demographics

The responses to this question provide a clear breakdown of the roles played in embedded systems projects.

- Hardware engineers are the largest group.
- Firmware and software engineers are also key players.
- Technical roles outweigh management roles on projects.



- Embedded systems design is the primary focus.
- Industrial and quality engineering are also significant.
- Engineering focuses reflect a blend of design and application.

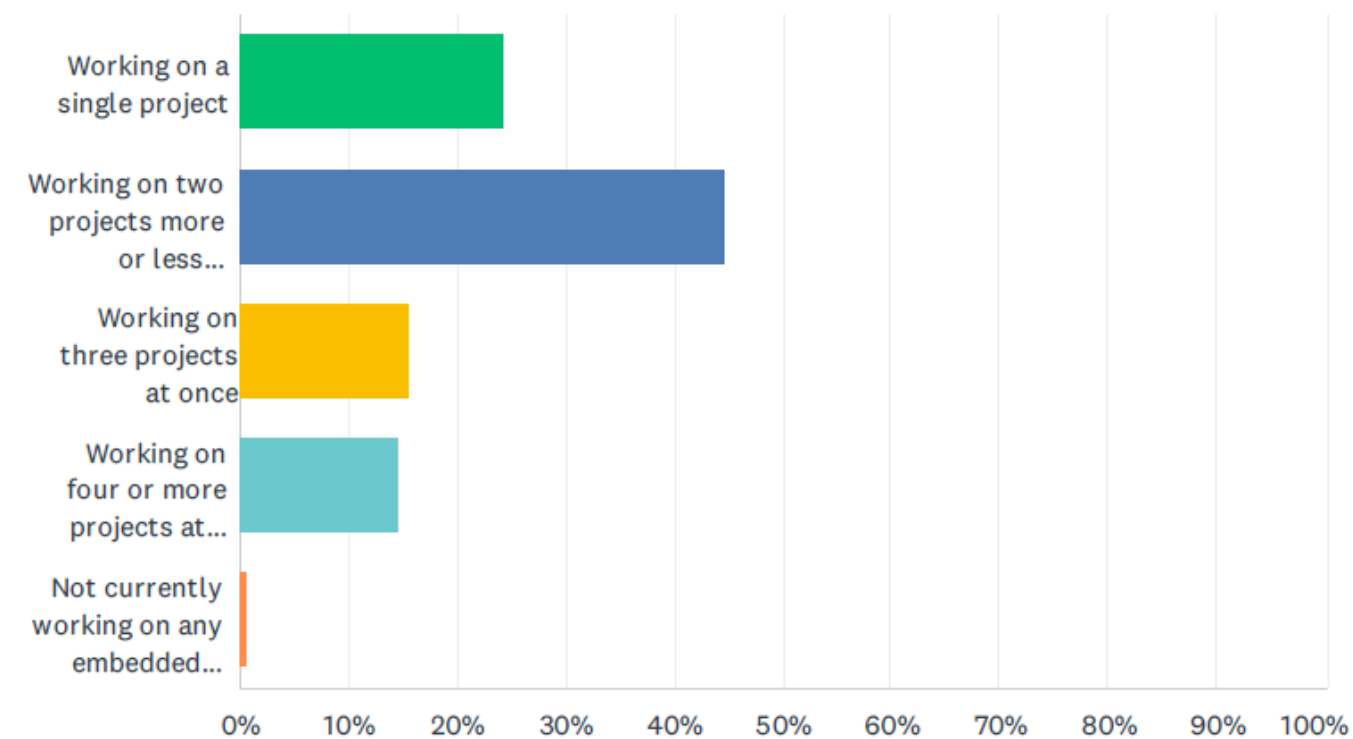


Project and Team

Project and Team Overview

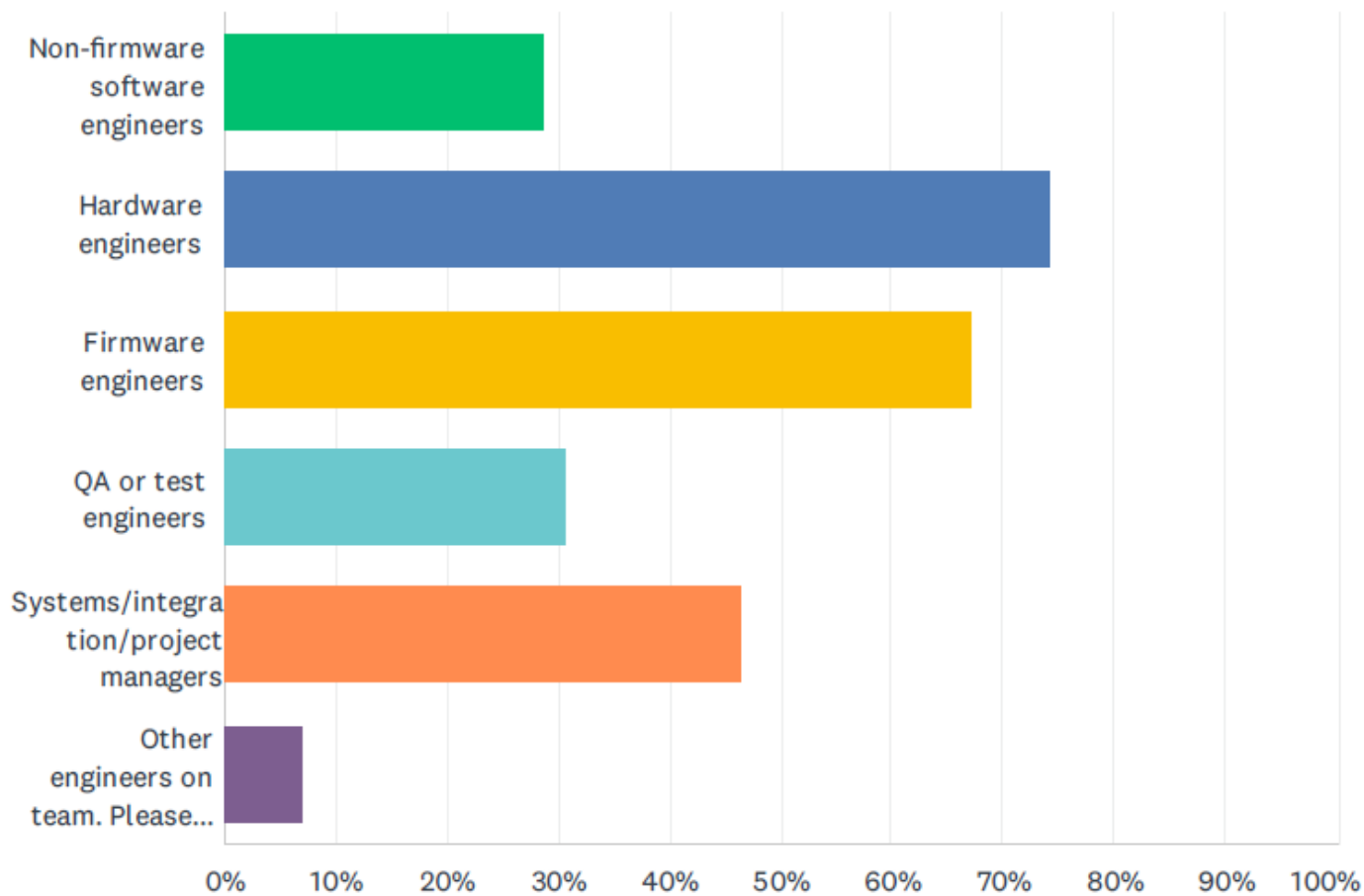
- ✓ Most projects are upgrades of existing designs.
- ✓ Two-thirds of projects are focused on improvements.
- ✓ Creating brand-new products is a smaller share of the work.

- ✓ Hardware engineers are the most common team members.
- ✓ Firmware engineers are also a core part of teams.
- ✓ Teams frequently include systems/integration/project managers.

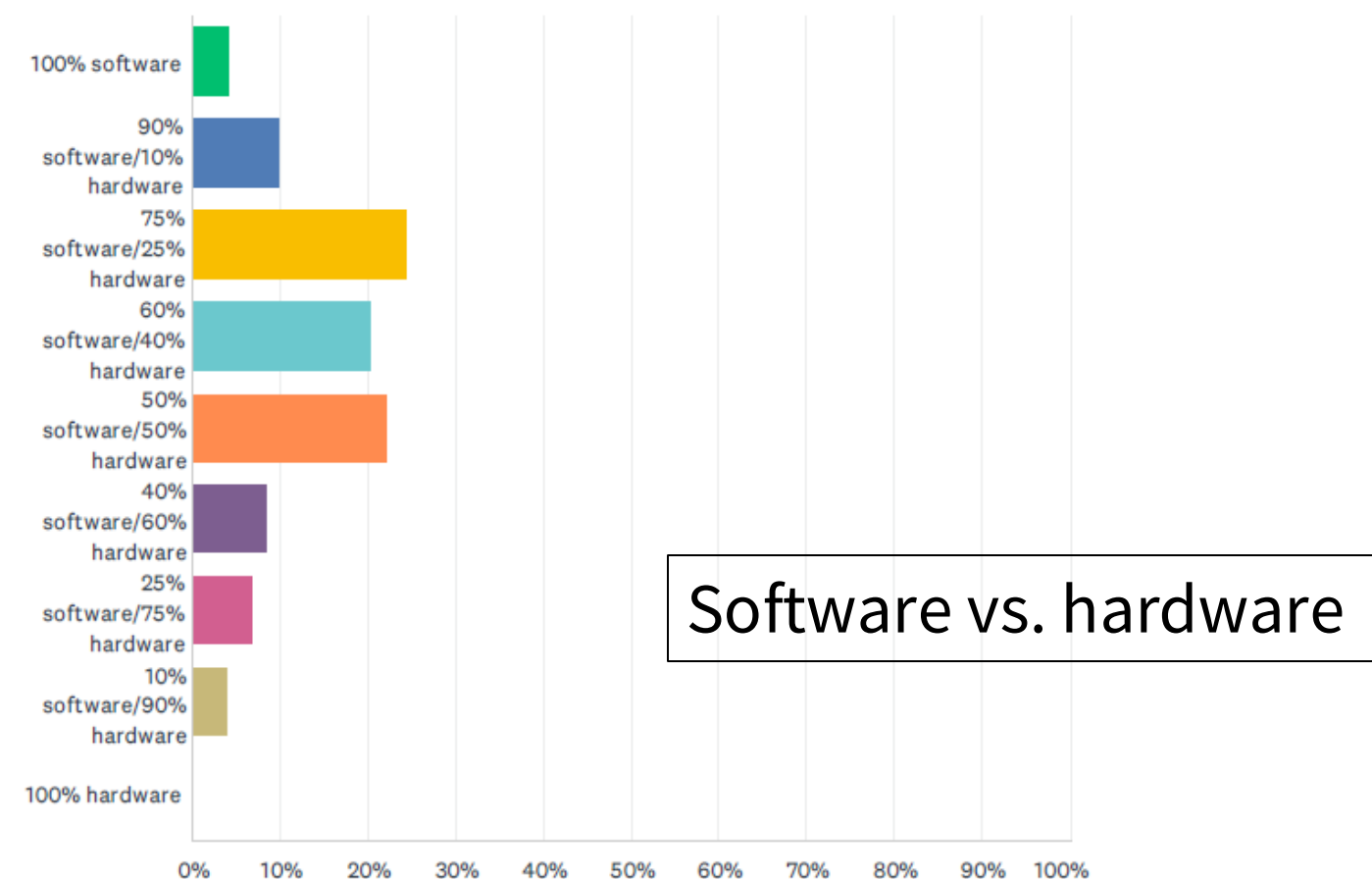


A large portion of the workforce is highly experienced:

30 years or more: 33.96%
20-30 years: 23.58%
15-20 years 11.56%



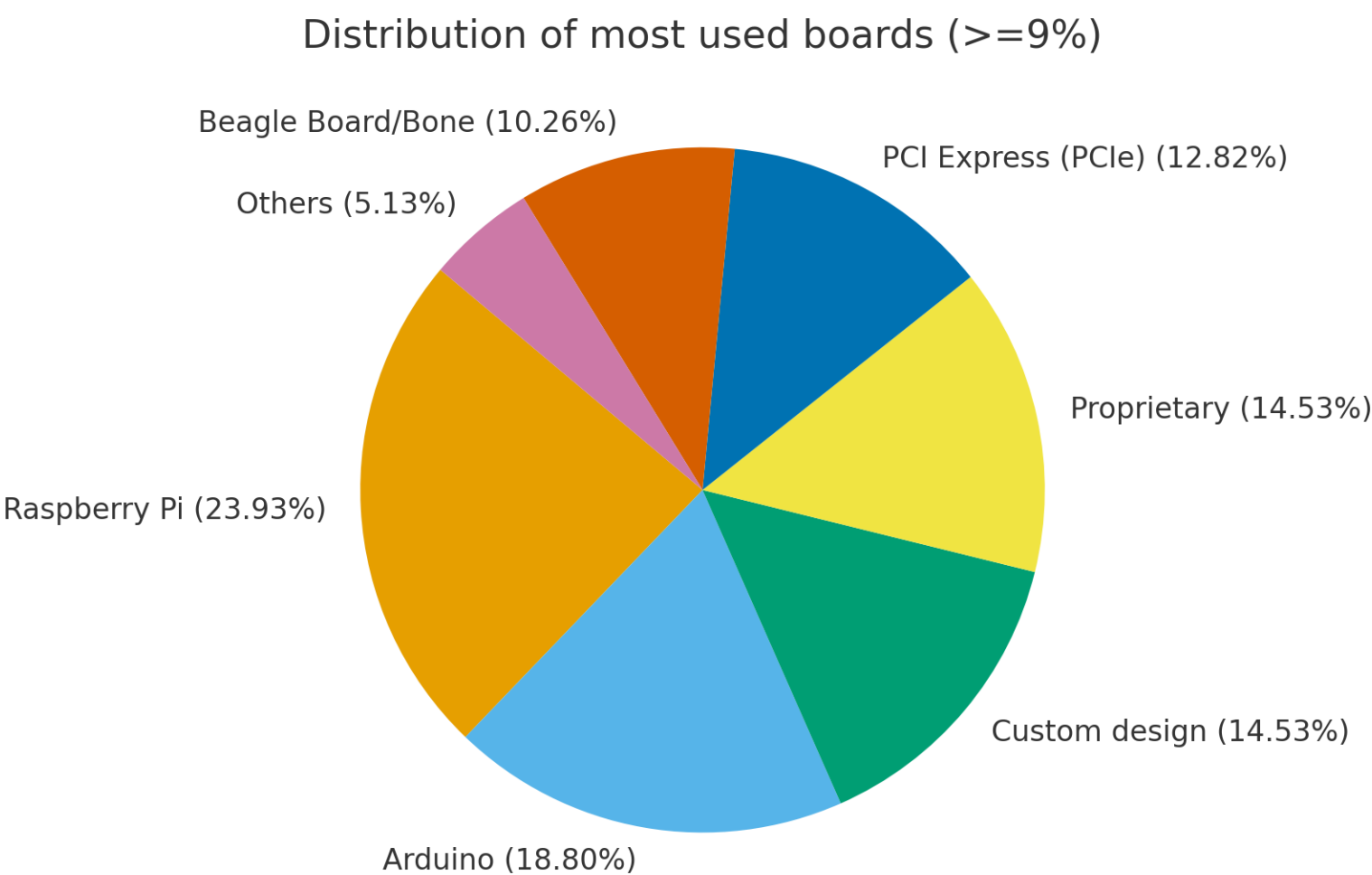
Hardware and Software Design



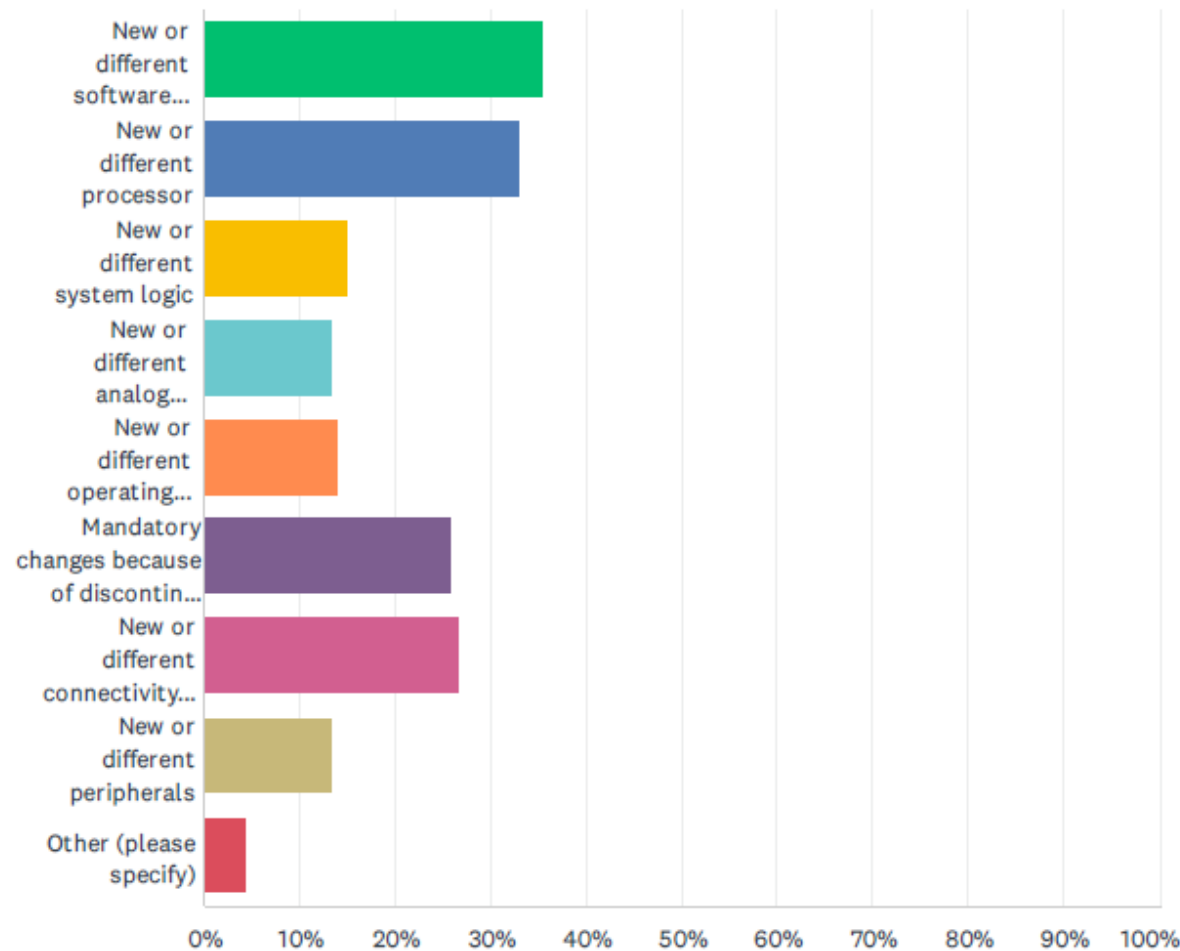
Software vs. hardware

The data shows a strong preference for custom-designed circuit boards in embedded systems development. The vast majority of respondents stated that they primarily build or subcontract their own boards.

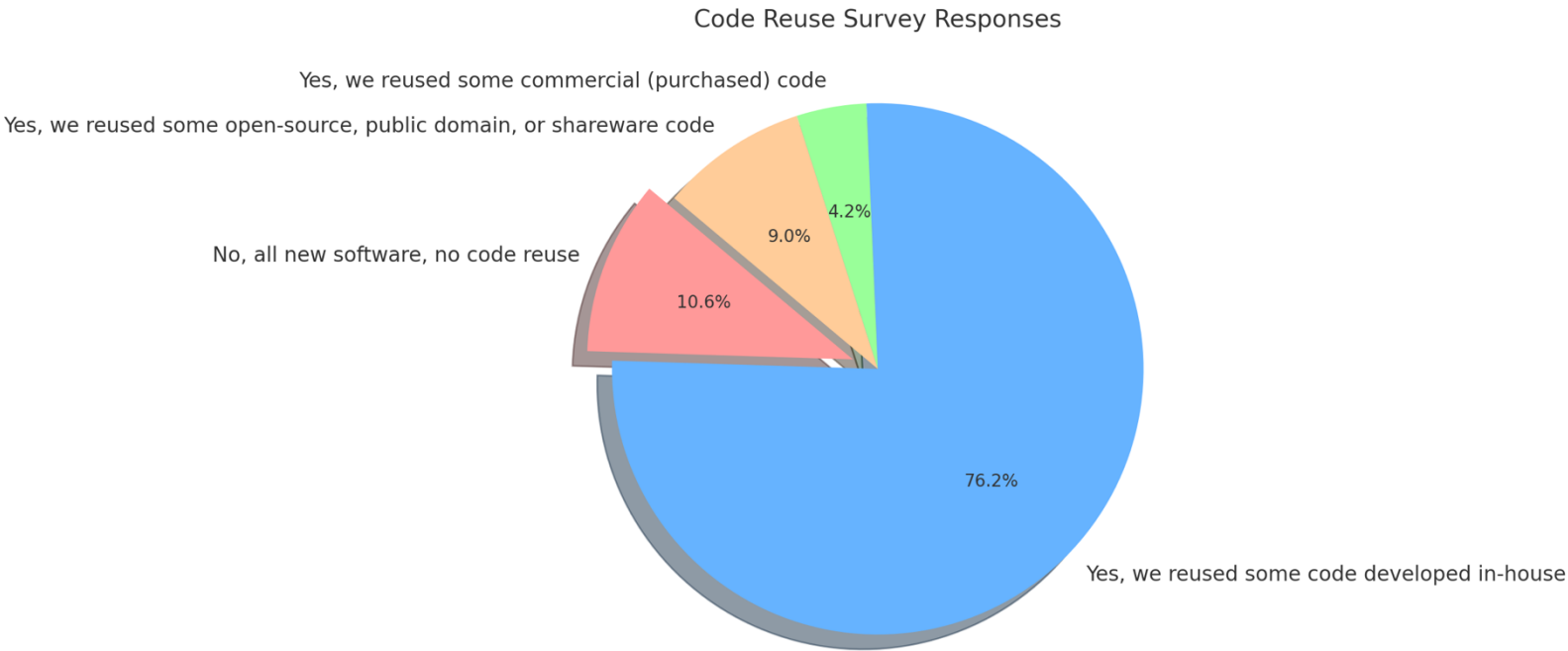
- ✓ Over 70% of projects begin with a development board.
- ✓ Using development boards is a standard practice.
- ✓ This approach helps accelerate the initial design phase.



Project Features & Reuse

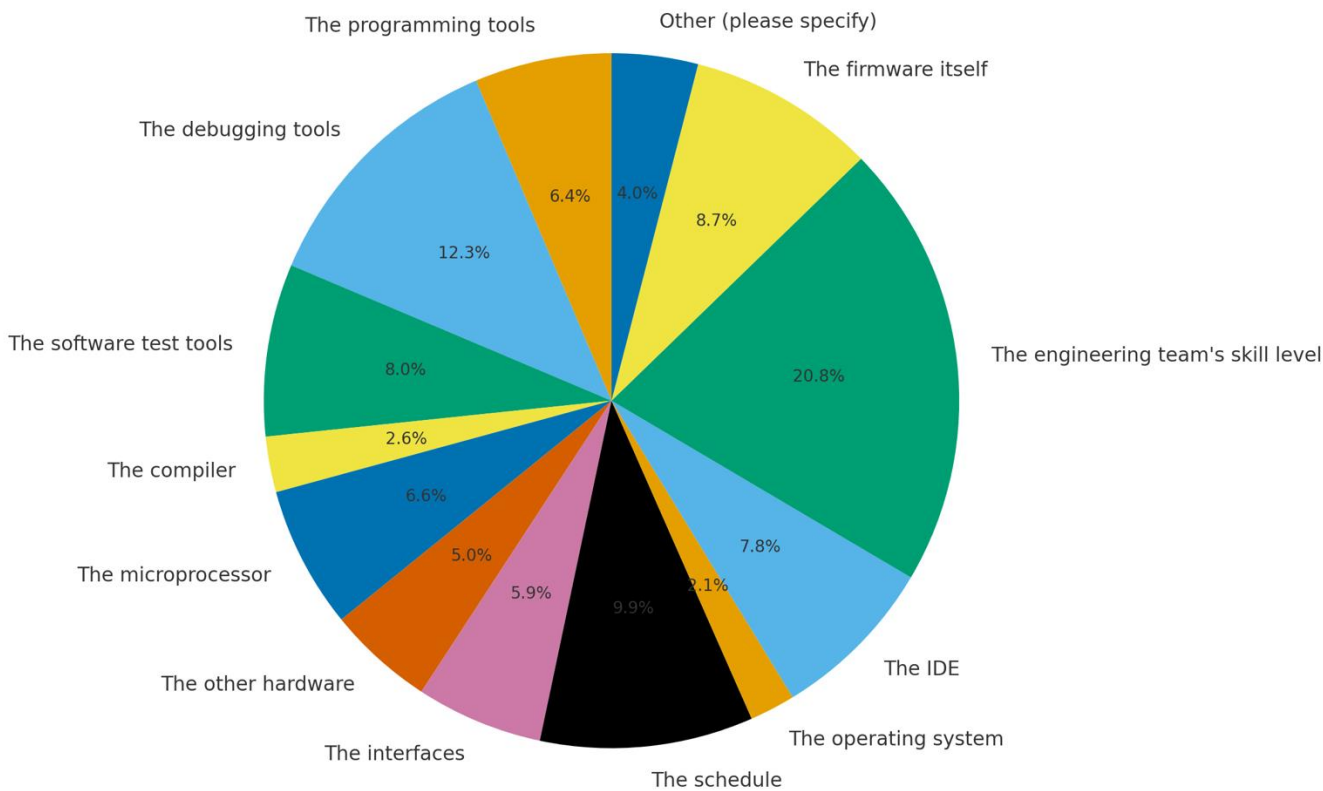


- ✓ New software features are the top motivator for upgrades.
- ✓ Switching processors is the second most common reason.
- ✓ Connectivity improvements are also a key factor.
- ✓ Real-time capability is the most included feature.
- ✓ Digital signal processing and wireless capability are also widely used.
- ✓ Projects require a blend of data processing and connectivity.



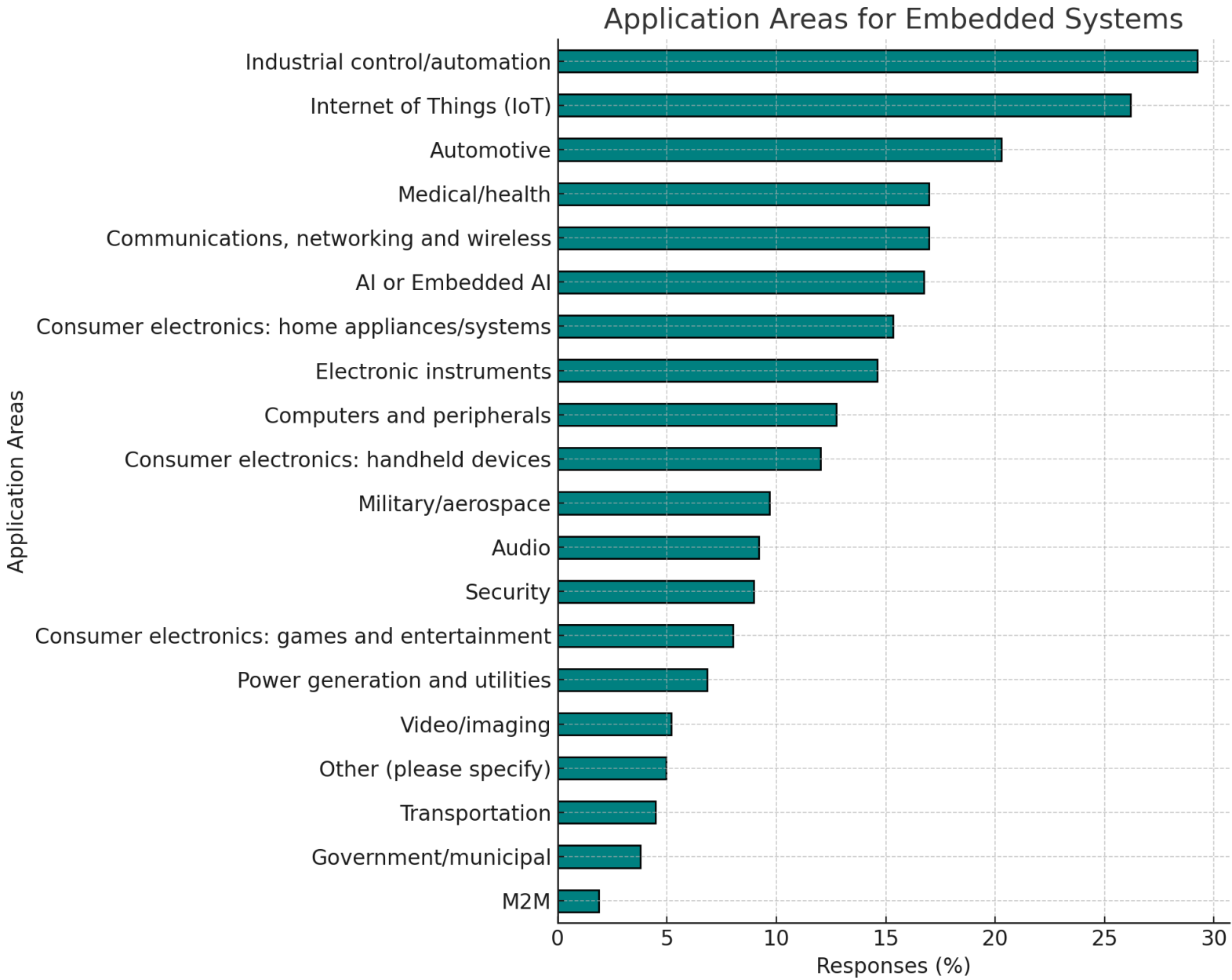
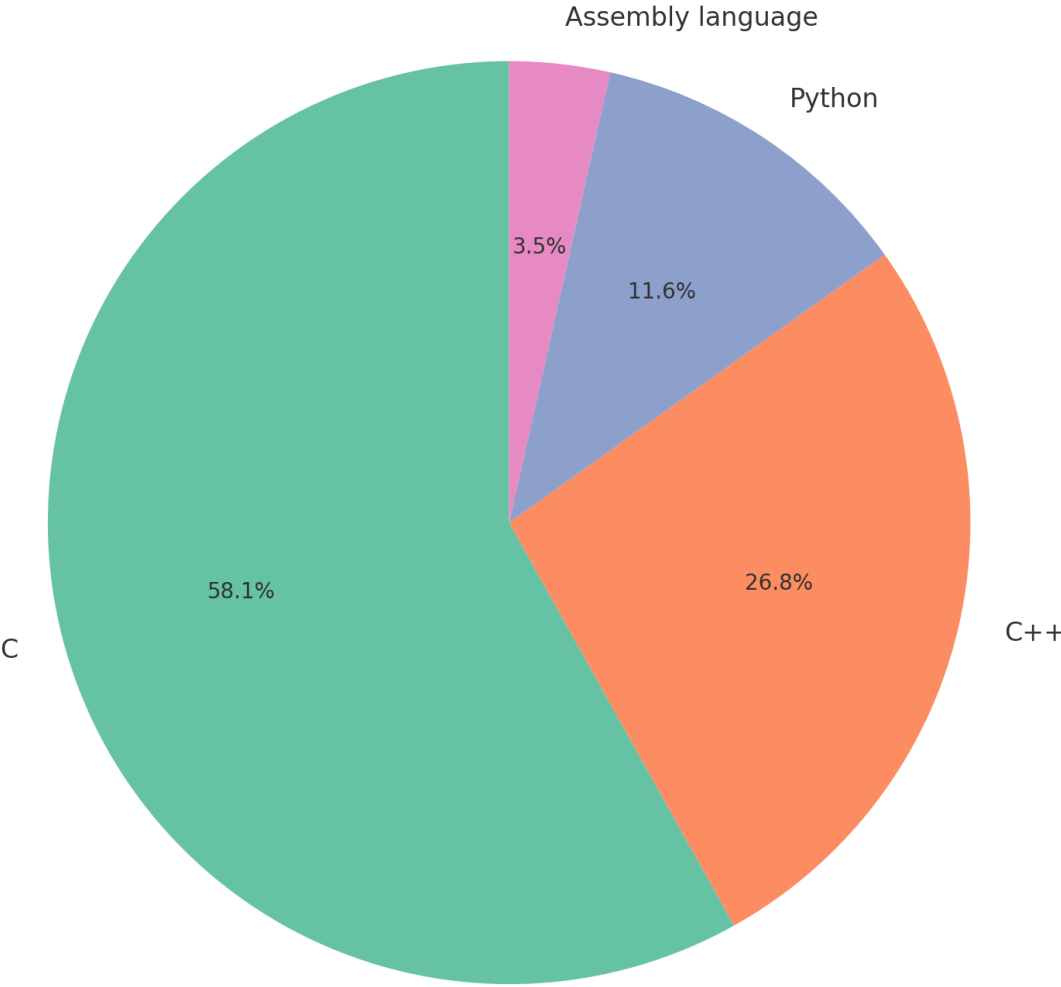
- ✓ Most projects reuse in-house developed hardware or IP.
- ✓ Over 65% of projects leverage existing hardware.
- ✓ Hardware reuse is a key strategy for efficiency.

If you could improve one of the following items in your current embedded project, which would it be?



Other Project Features

Top 4 Programming Languages in Current Use



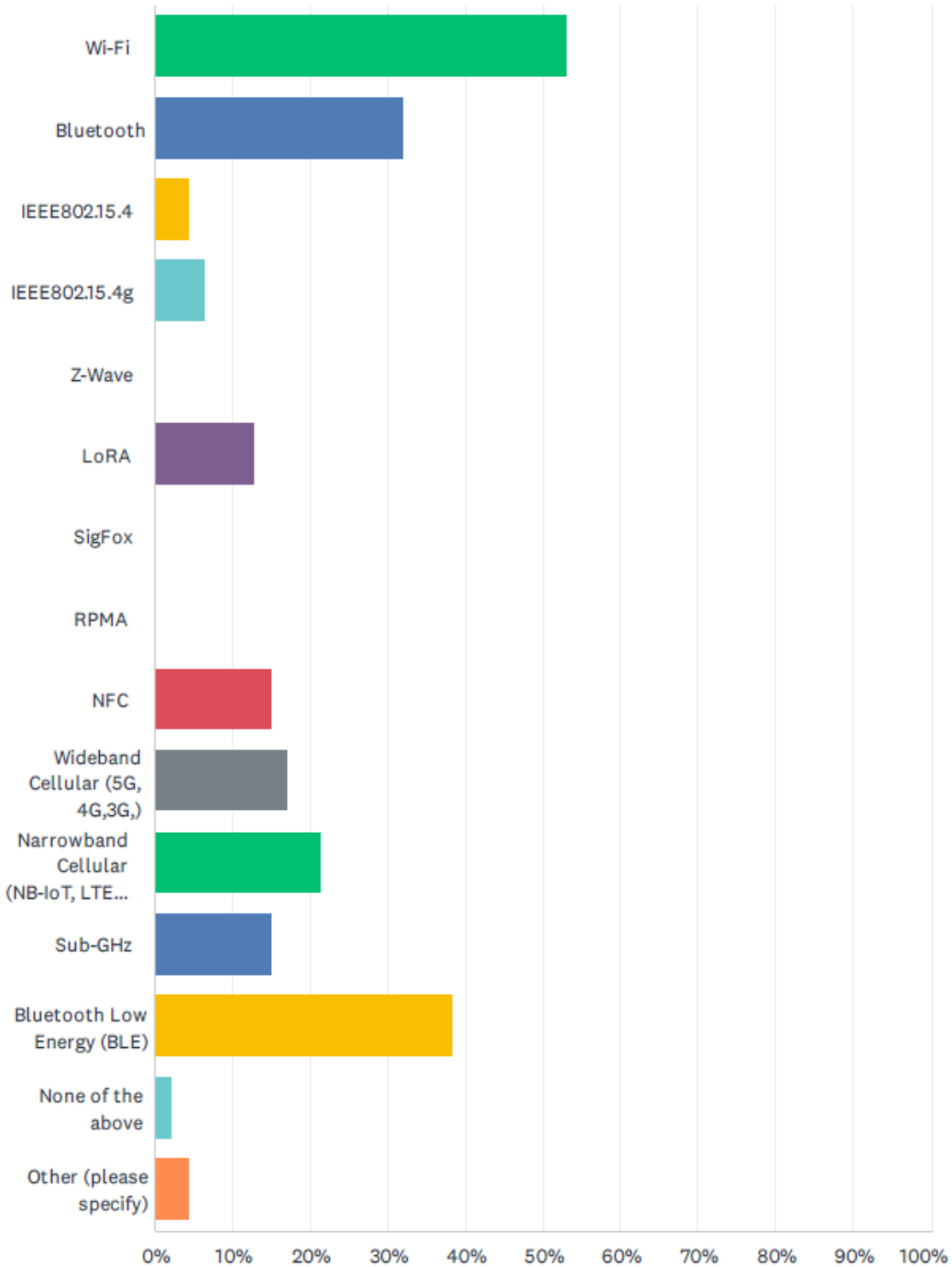
- ✓ Industrial control and automation is the top application area.
- ✓ IoT is the second most common application.
- ✓ Automotive is the third most common application.

Technologies and Challenges

Wireless & Advanced Technologies

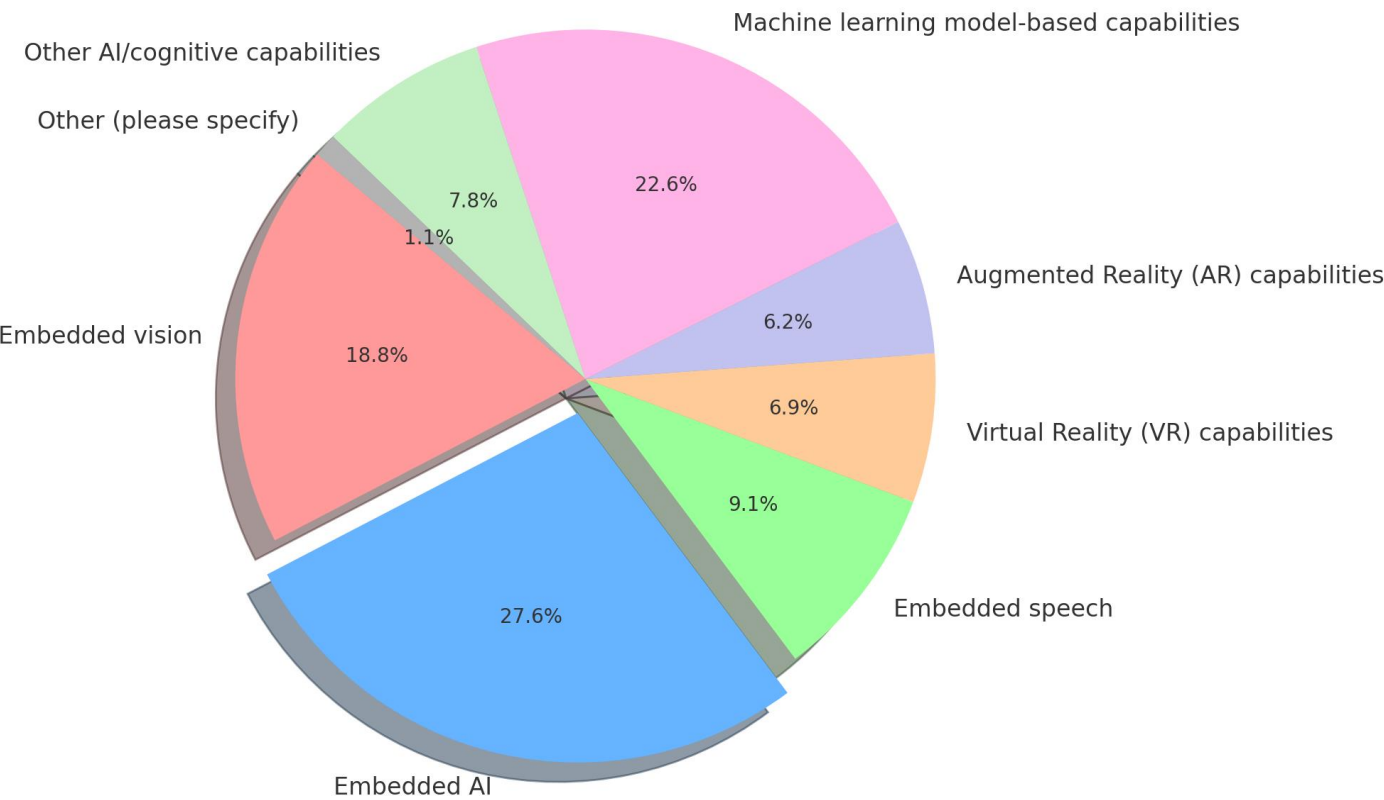
Protocols/Stack

- ✓ Wireless I/O Link and BLE mesh are the most used.
- ✓ ZigBee is the next most popular protocol.



Technologies

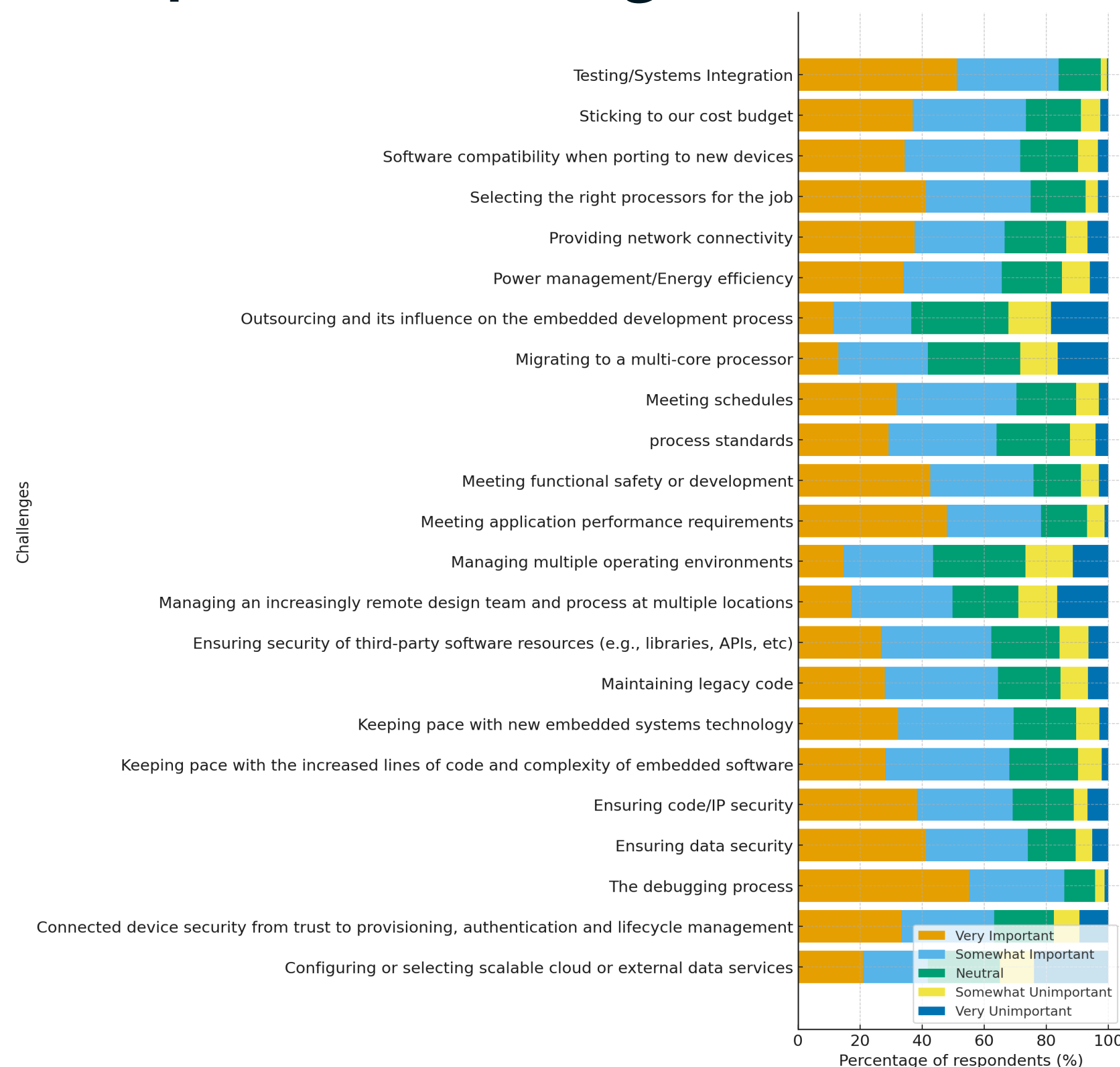
Currently in Use



Future

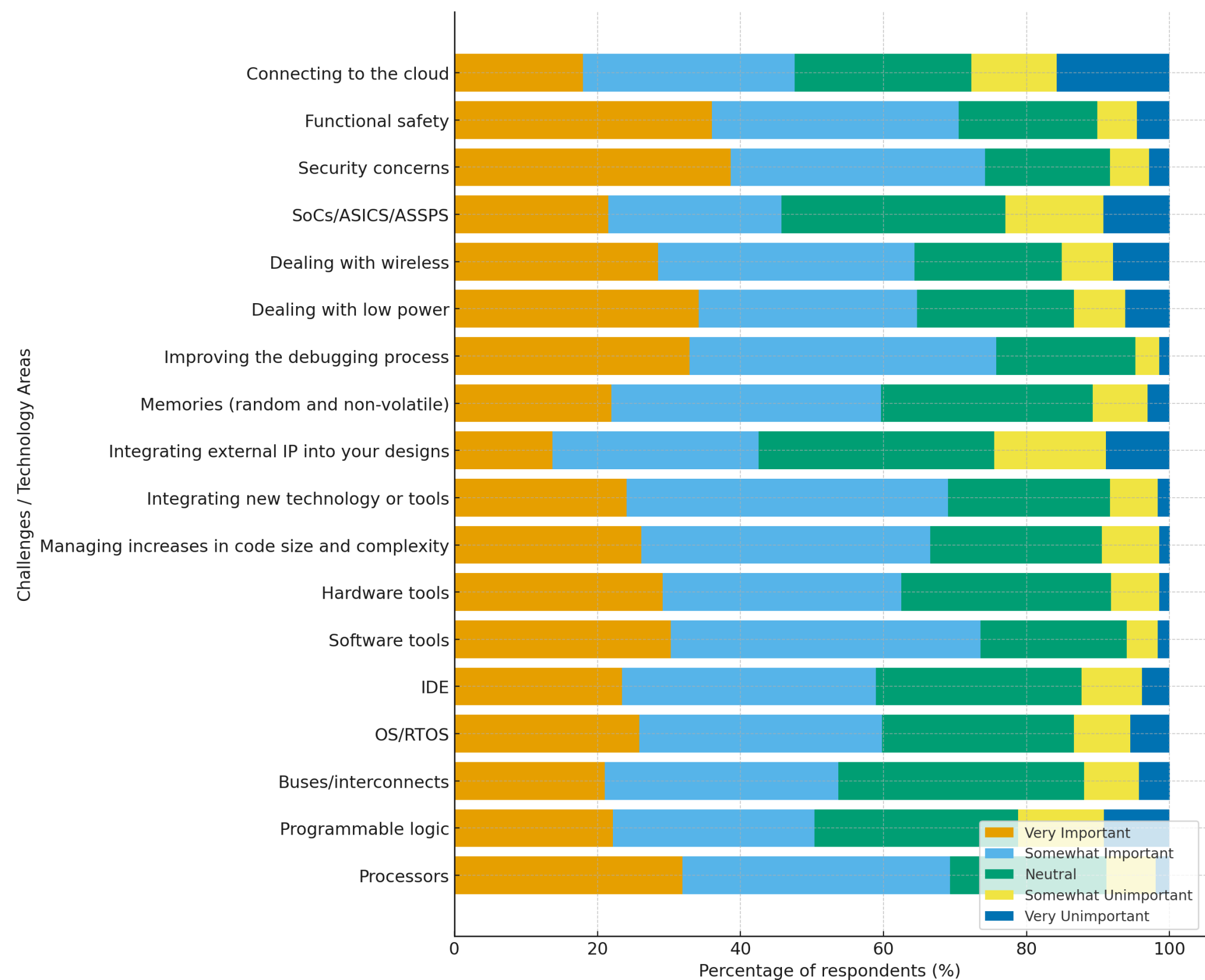
- ✓ More people plan to use advanced technologies in the future.
- ✓ Embedded AI is the most considered future technology.
- ✓ Machine learning is the second most considered technology.

Development Challenges



- ✓ Top challenges: Testing, Debugging, Performance
- ✓ Security: data, code/IP, third-party SW, device lifecycle
- ✓ Rising concerns: complexity, keeping up with tech
- ✓ Standards/safety: important but market-dependent
- ✓ Lower urgency: outsourcing, remote teams, energy efficiency

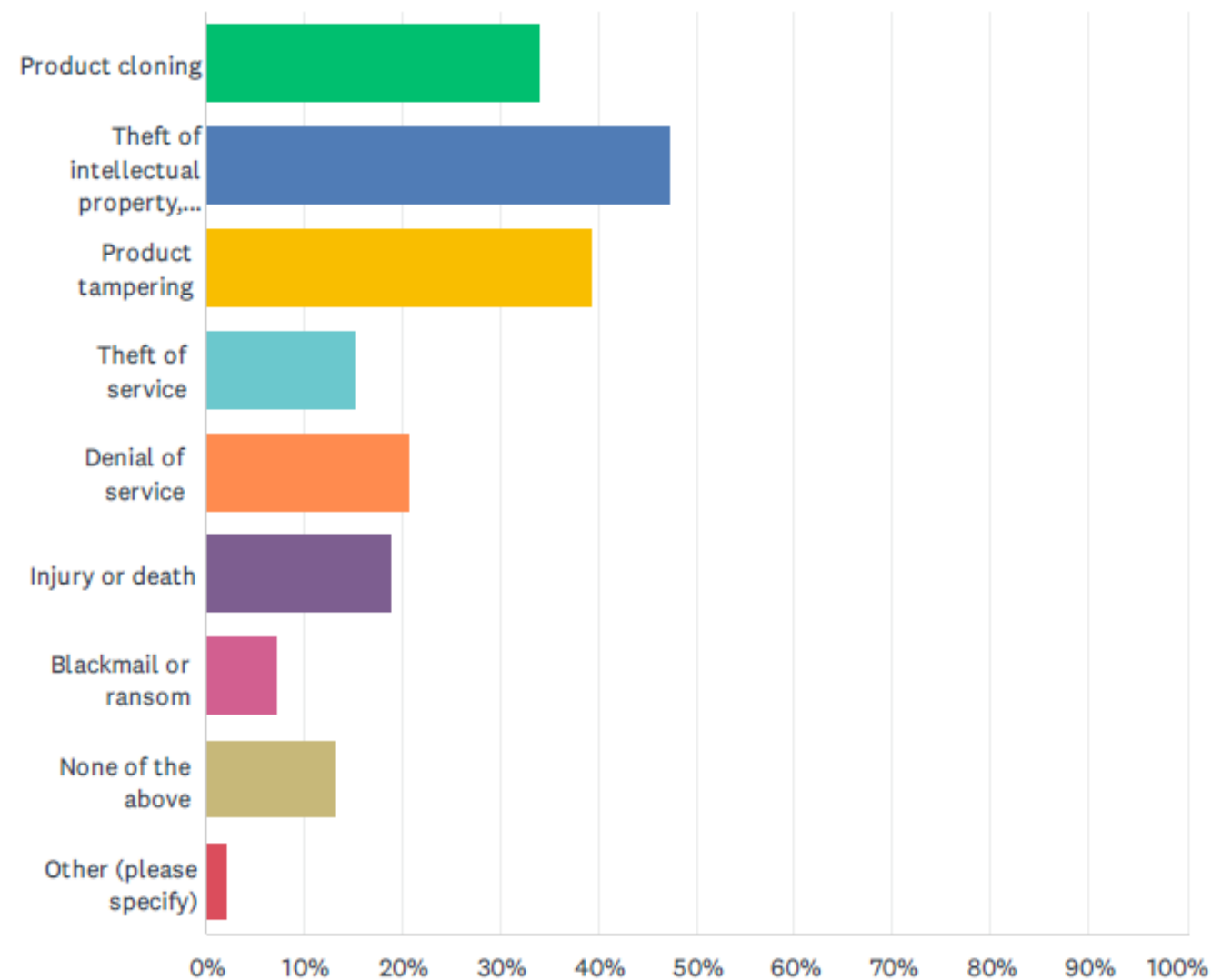
Future Challenges



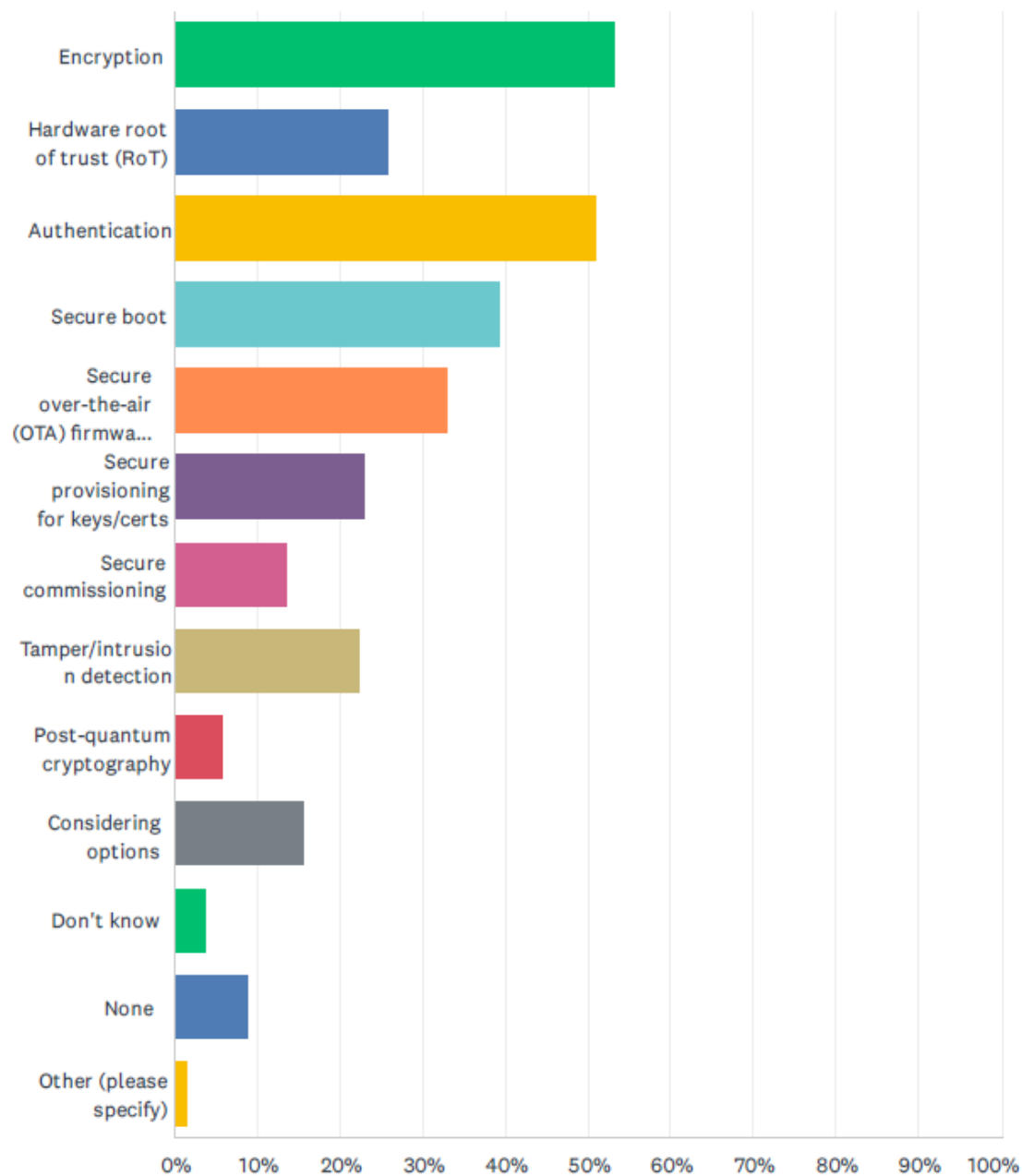
- ✓ Critical hurdles: Security, Safety, Low Power
- ✓ Tooling matters: Debugging, Software/Hardware tools, IDEs, OS/RTOS
- ✓ Foundational but mature: Processors, Programmable Logic
- ✓ Growing concerns: Complexity, Integrating new tech
- ✓ Lower priority: Cloud, External IP

Security

Encryption and Authentication are the most widely used methods, highlighting a strong focus on protecting data and verifying device identity.



The survey reveals that the most significant security concerns for embedded projects are theft of intellectual property, customer privacy violations, or theft of data, followed by product tampering and product cloning.

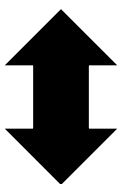


Hardware support for security

MCU security features: 49.53%

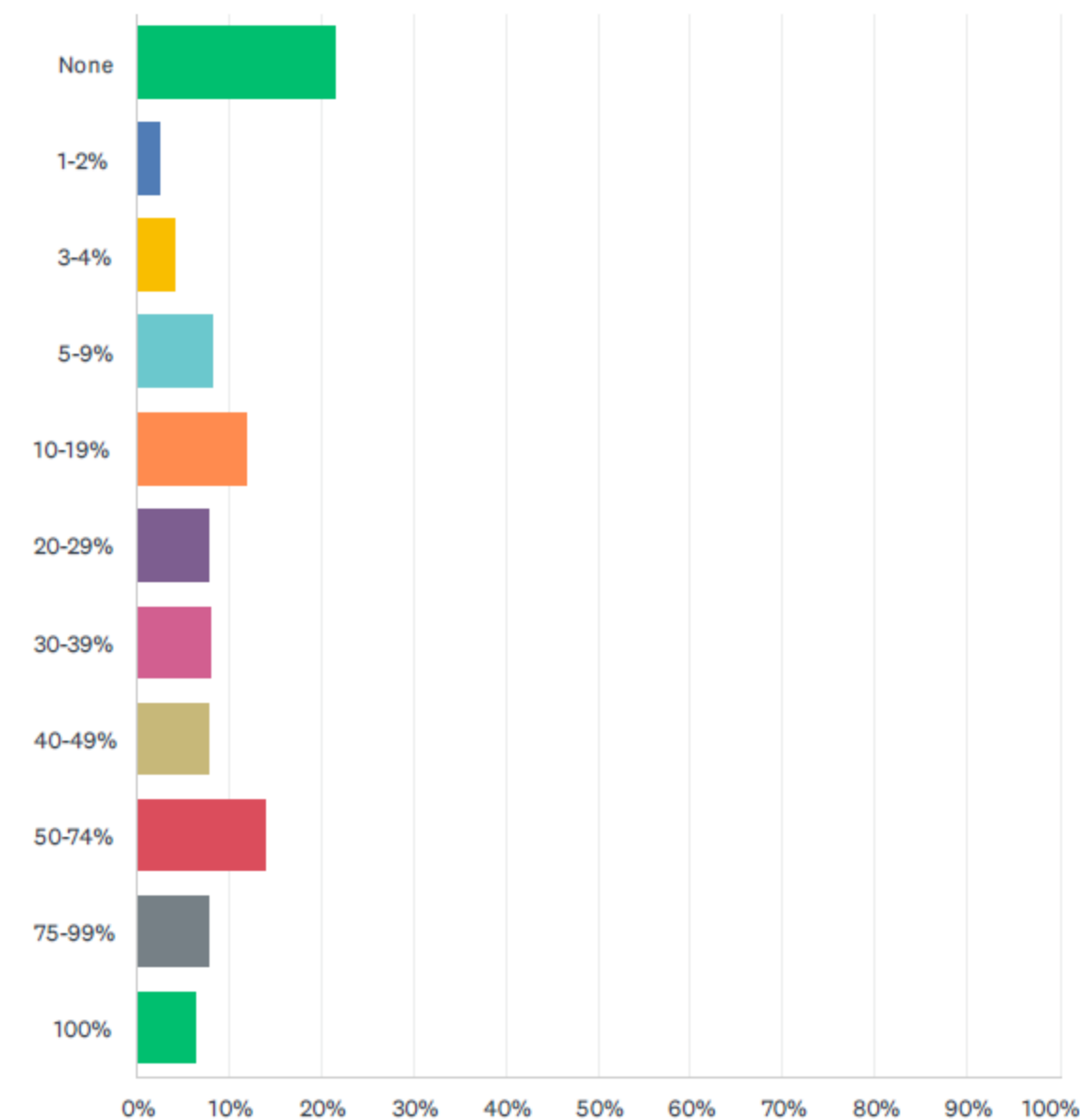
Hardware security module: 29.95%

Software-only security: 28.54%

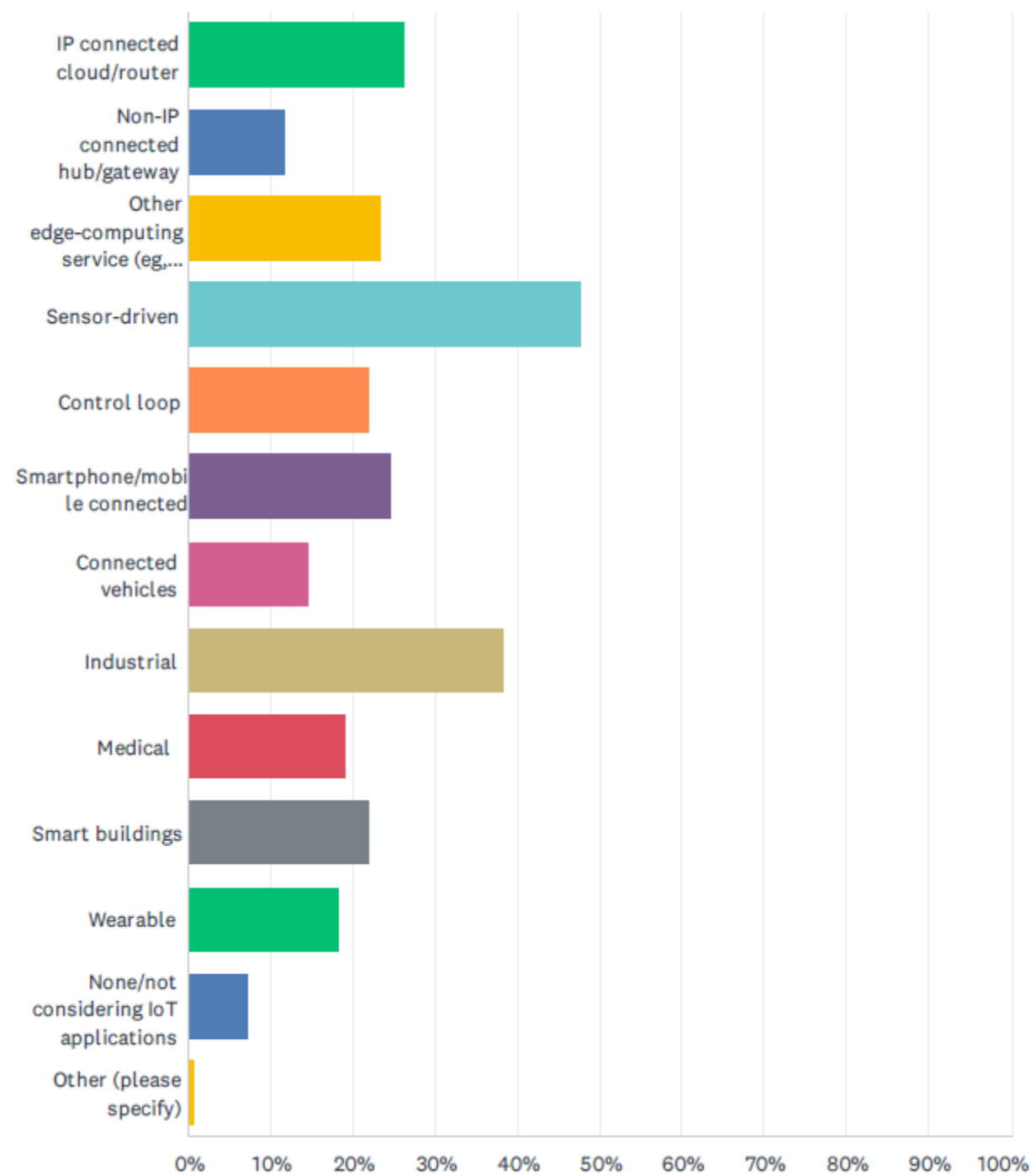


The survey highlights that most professionals rely on MCU security features. This indicates a trend towards using the security capabilities already integrated into the microcontroller itself.

IoT

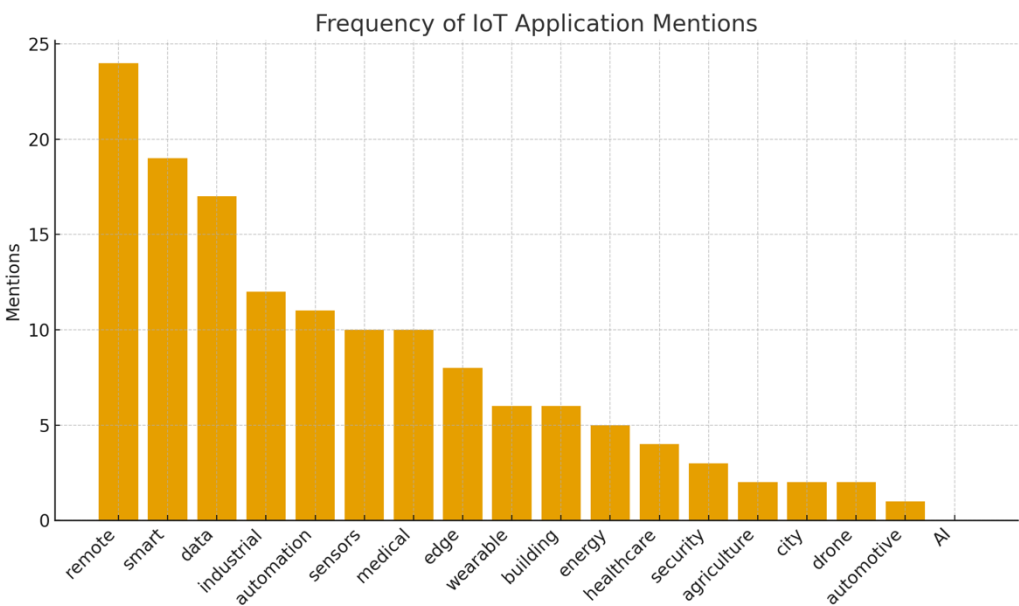


While over a fifth of respondents stated they will have no IoT projects, a significant portion of the remaining projects are highly dedicated to IoT.



The survey shows that sensor-driven applications are the most common type of IoT project.

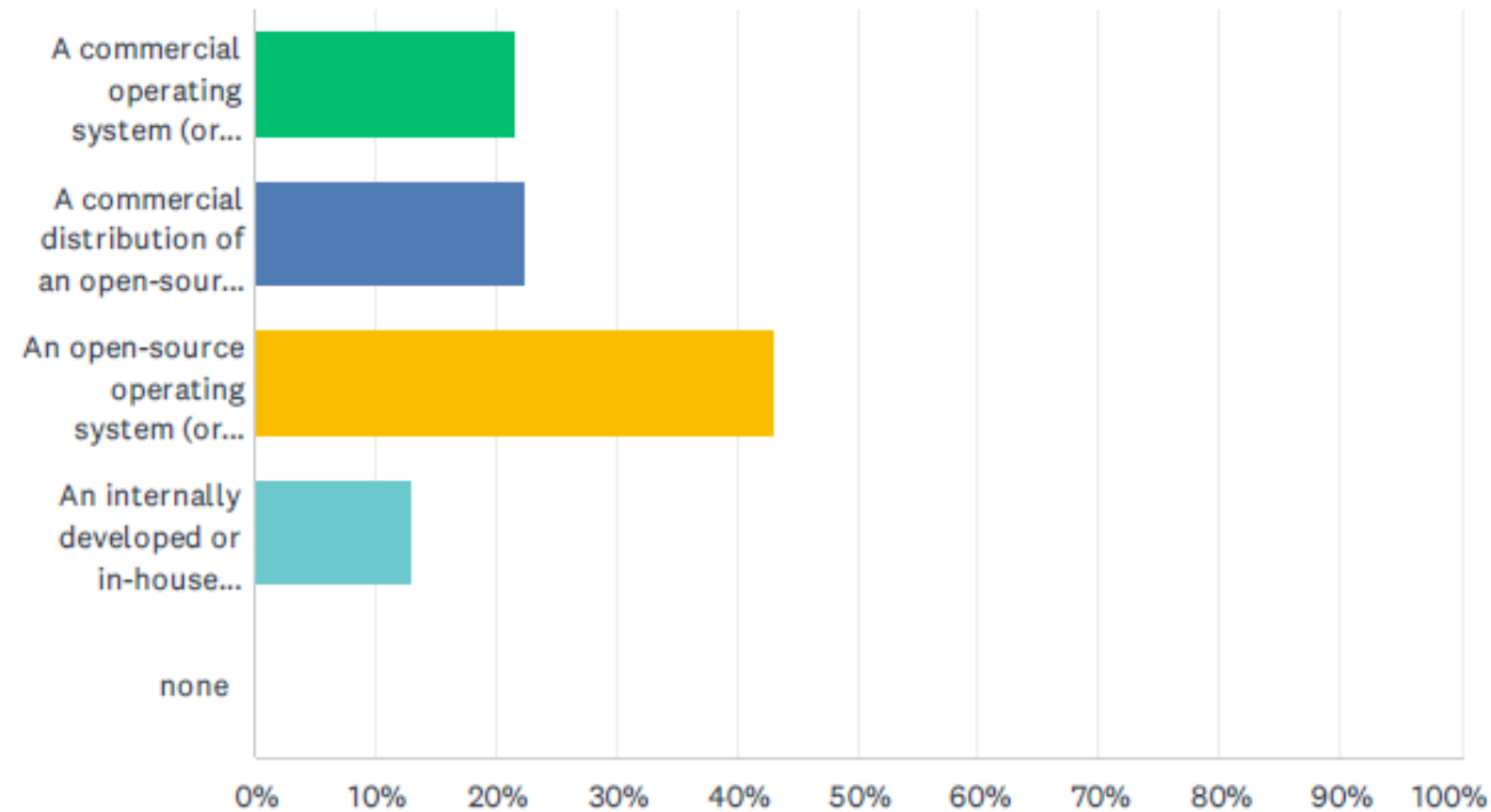
Considering all applications of which you are aware, what do you regard as the most interesting use of the IoT?



- ✓ Industrial & automation dominate: strong focus on predictive maintenance, process optimization, and sensor-driven efficiency.
- ✓ Healthcare & wearables stand out as impactful: enabling continuous monitoring, remote care, and life-saving insights.
- ✓ AI, edge computing & security are seen as key enablers: powering real-time decision-making while ensuring safe and scalable IoT connectivity.

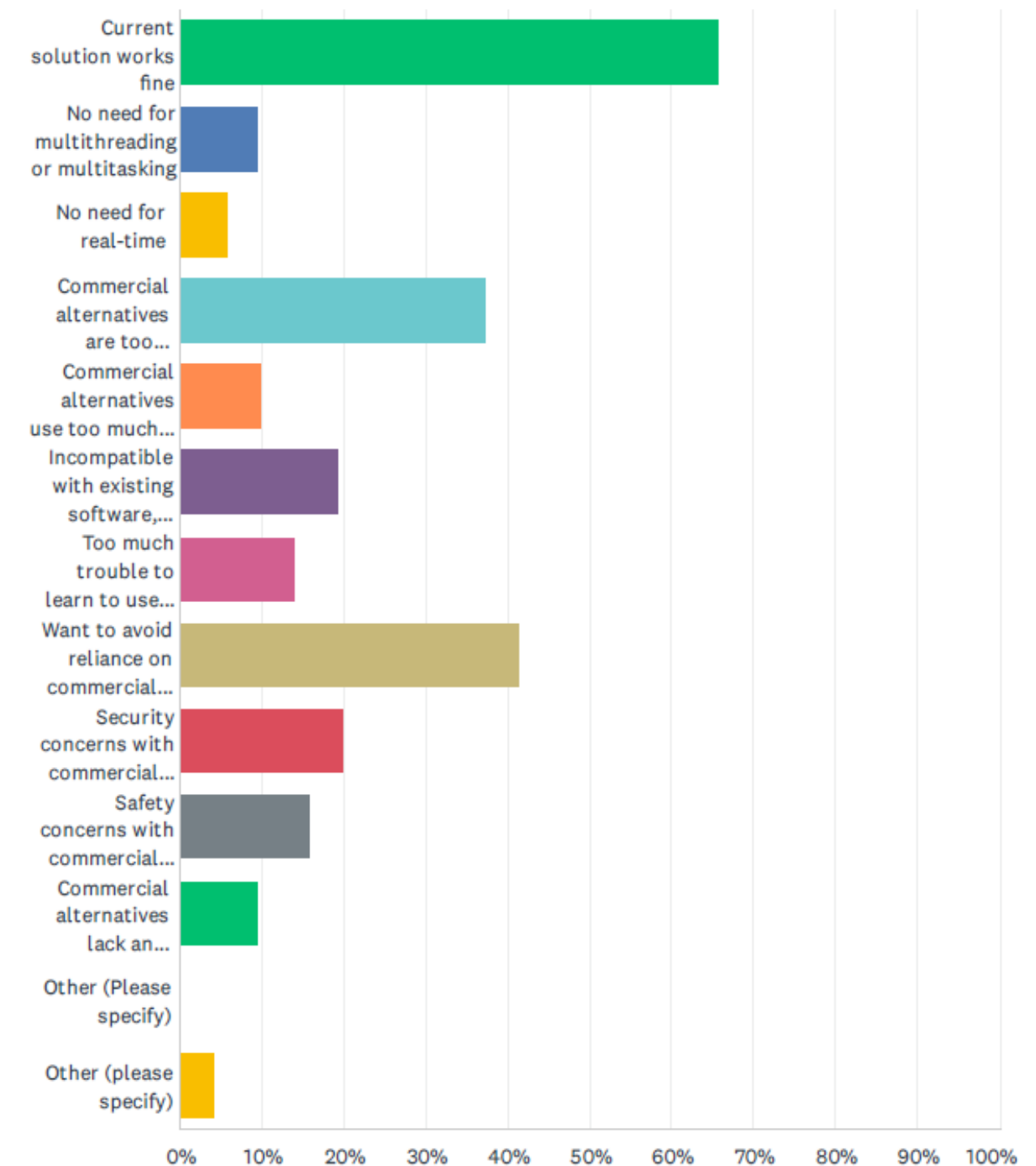
Operating Systems, Virtualization and Processors

Operating Systems (Part 1)



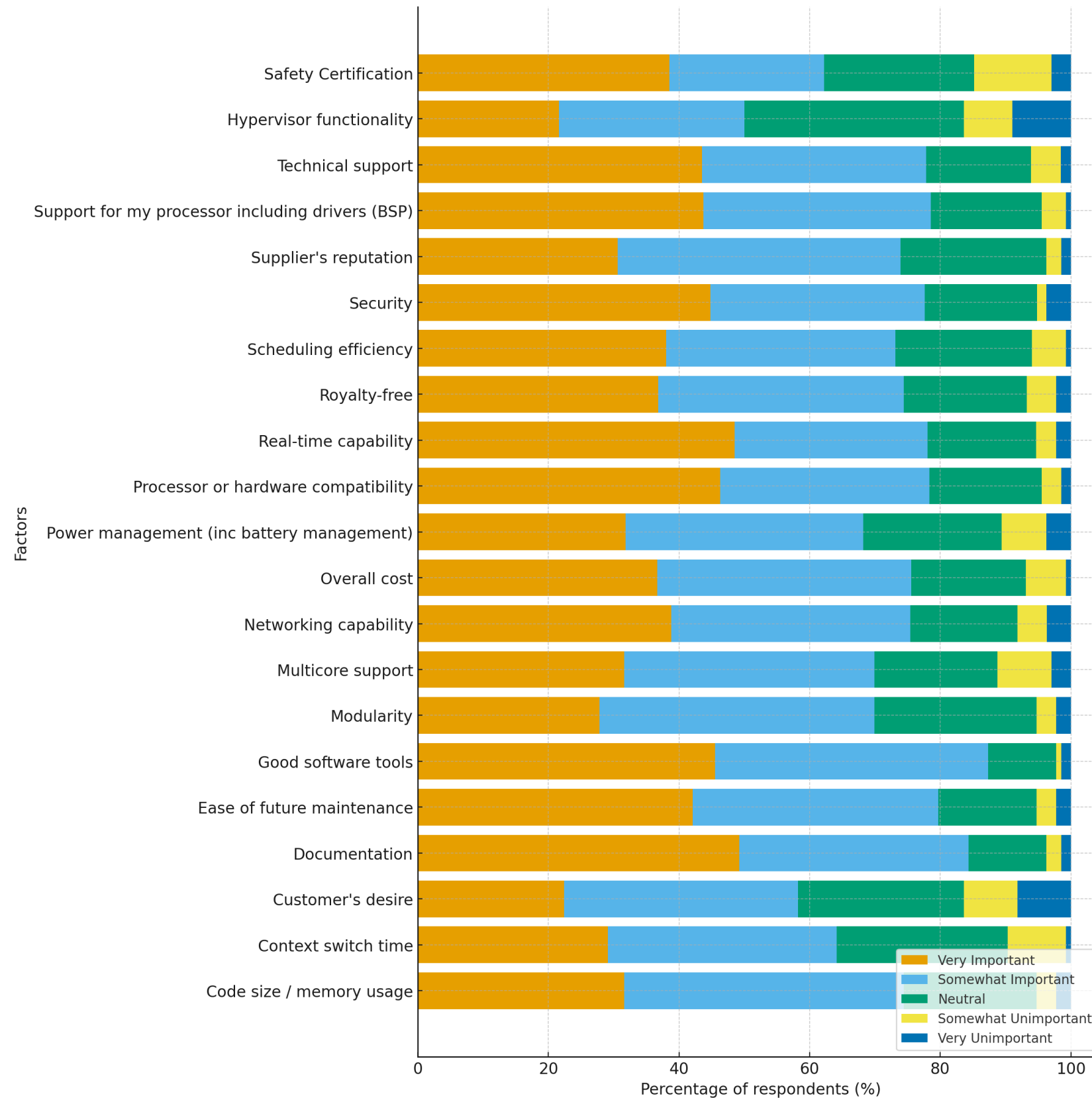
- ✓ Open-source operating systems are the most common choice.
- ✓ Commercial support for open-source systems is the next choice.
- ✓ Customization and cost are key drivers for OS choice.

The data shows that most embedded projects use some form of operating system or scheduler. Over 70% of respondents answered yes, confirming that these systems are a fundamental part of modern embedded design.

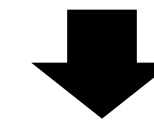


For those who do not use a commercial operating system, the top two reasons are that the current solution works fine, and they want to avoid reliance on a commercial supplier.

Operating Systems (Part 2)



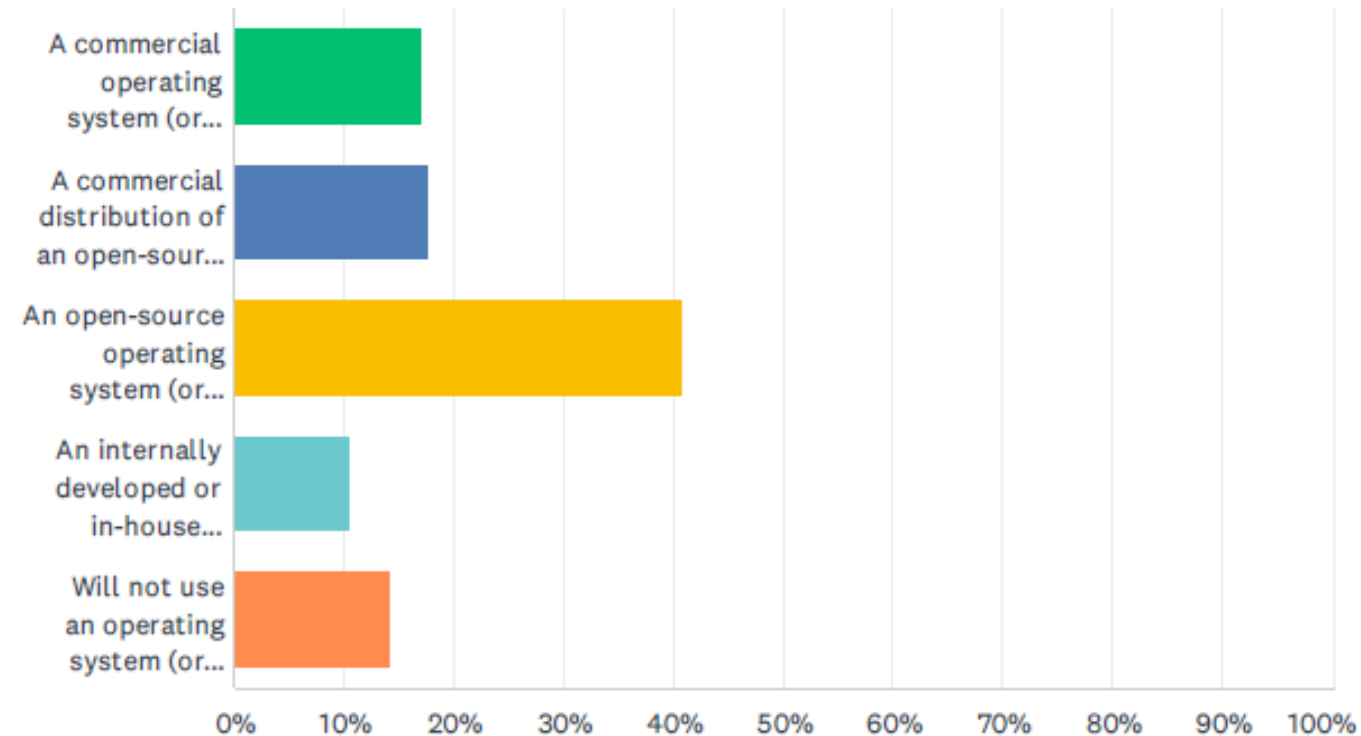
How important are the following factors regarding your decision to use a commercial operating system, RTOS, kernel, etc.?



- ✓ Top priorities: Documentation, software tools, real-time capability, processor compatibility, and security
- ✓ Strong support needs: Technical support, ease of maintenance, supplier reputation
- ✓ Context-specific factors: Hypervisor functionality, customer's desire, and cost seen as less critical overall

Operating Systems (Part 3)

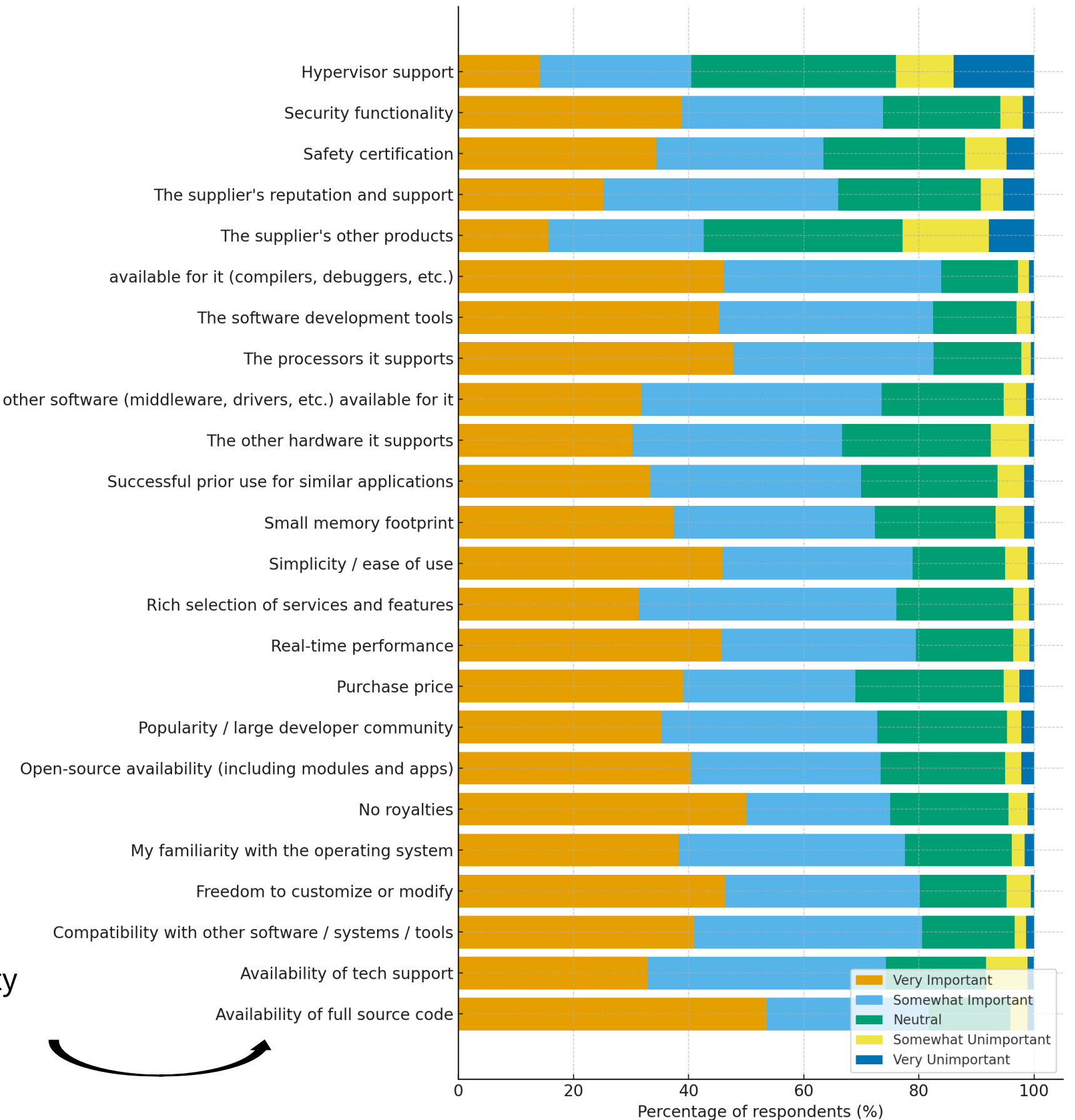
Looking ahead, the survey indicates that an open-source operating system without commercial support will continue to be the dominant choice for future embedded projects.



When choosing an operating system, the most critical factor for developers is the availability of full source code.

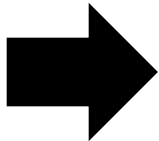
- ✓ Critical factors: Source code access, no royalties, customization, processor support, security
- ✓ Strong enablers: Ease of use, real-time performance, developer tools, compatibility
- ✓ Lower priority: Supplier reputation, vendor's other products, hypervisor support

OS Selection Factors

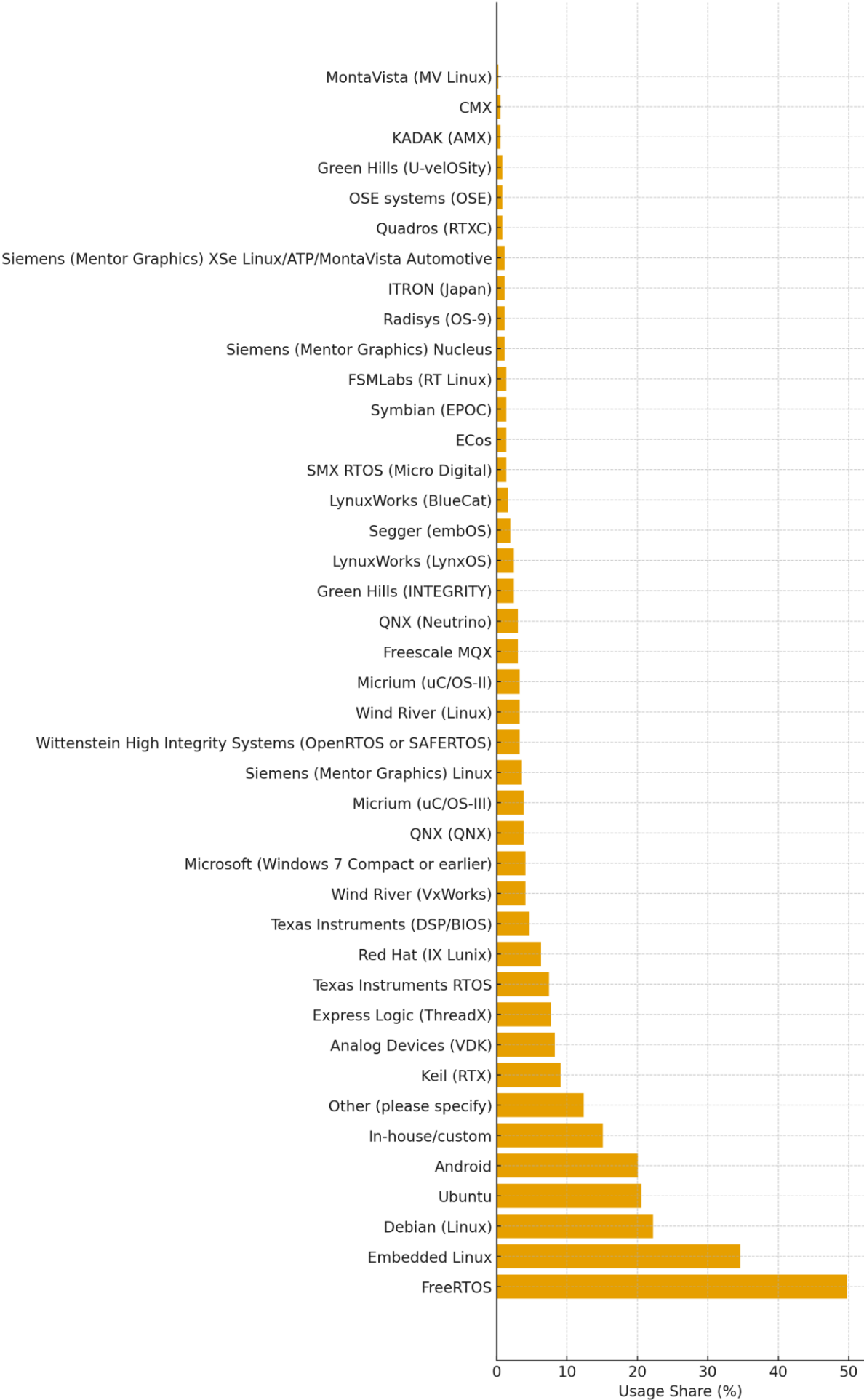
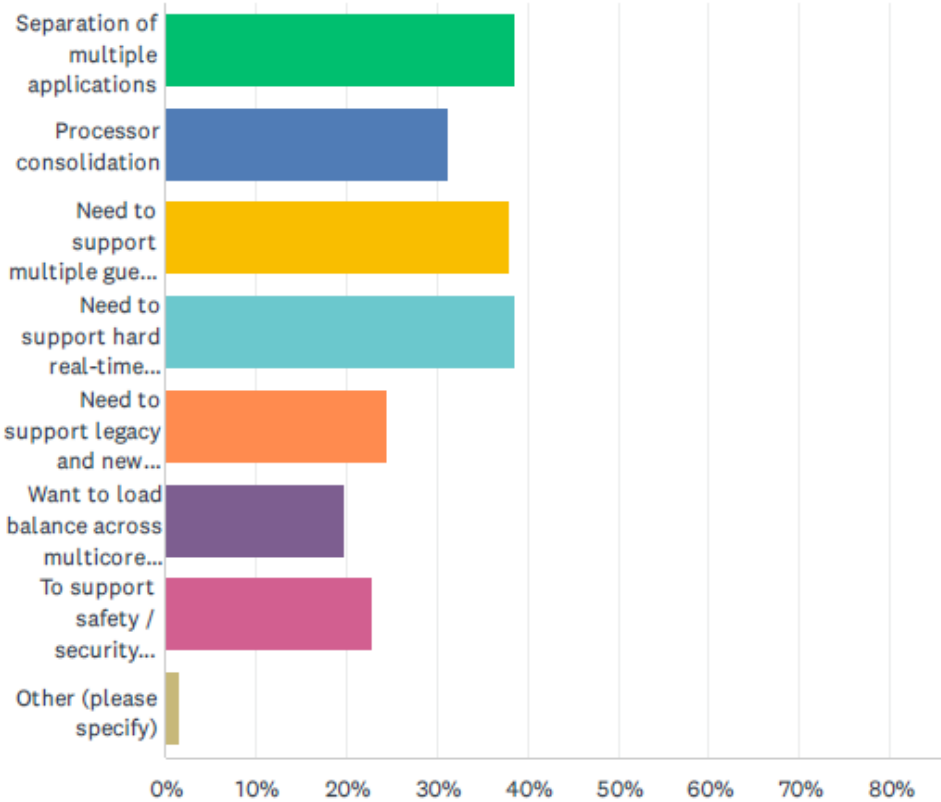
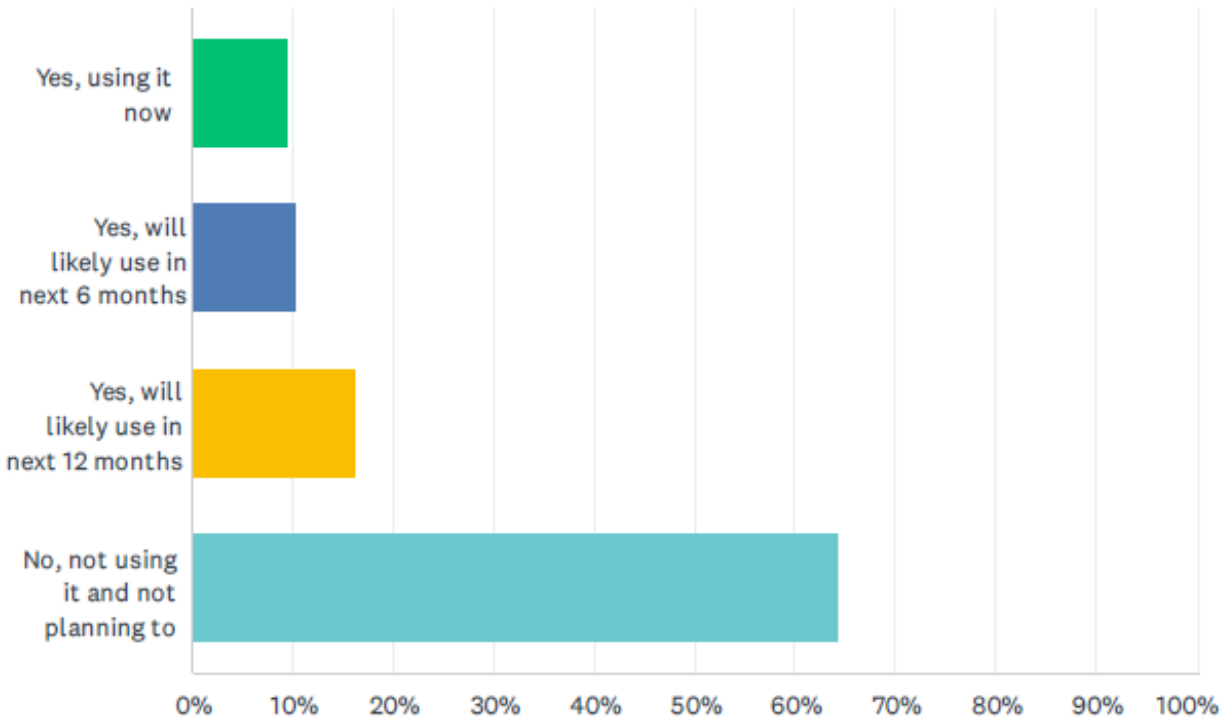
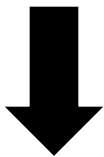


OS & Virtualization Vendors

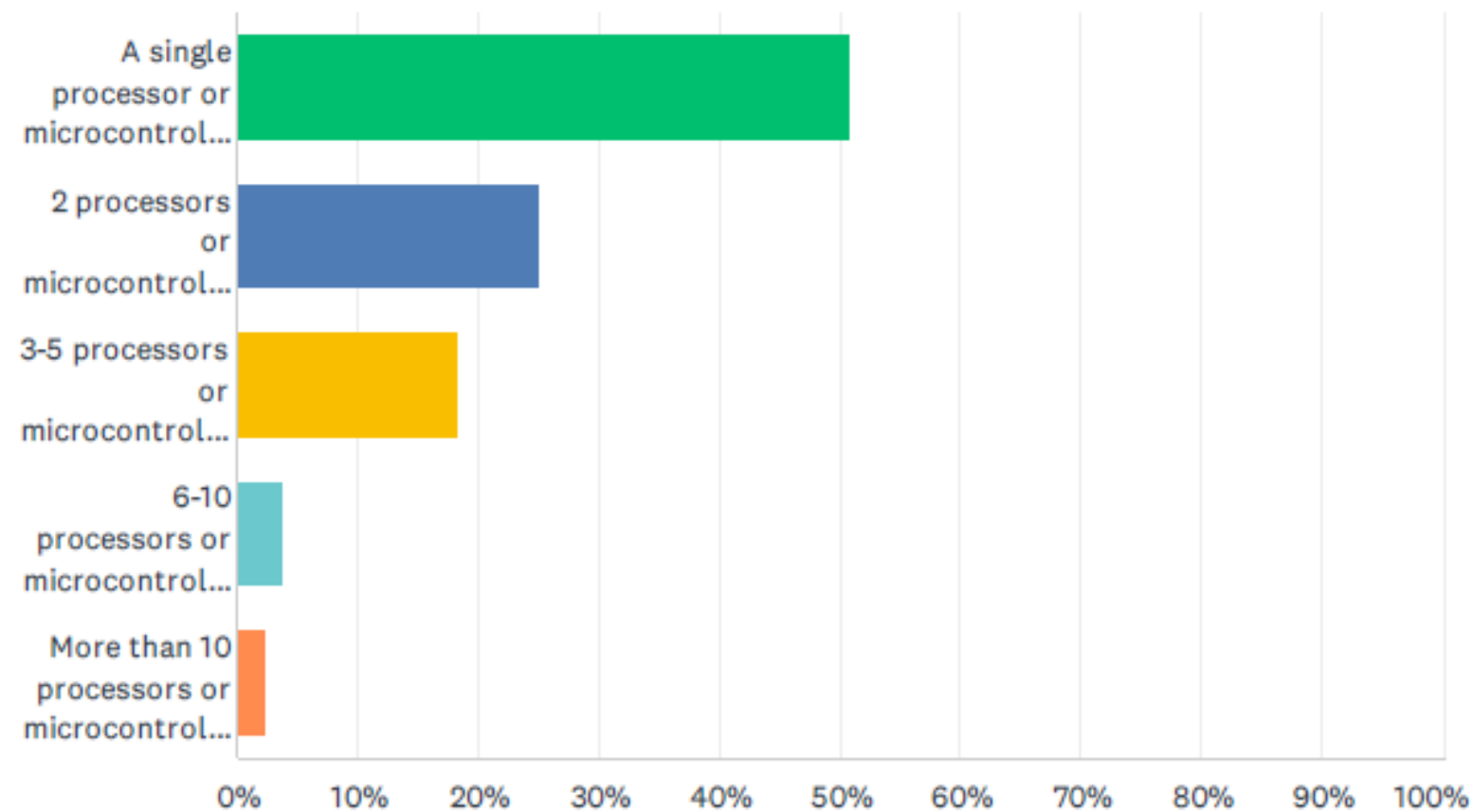
Please select the operating systems you are currently using or considering using in the next 12 months for a commercial product development project. (Only include non-RTOS operating systems that you embed into your projects.)



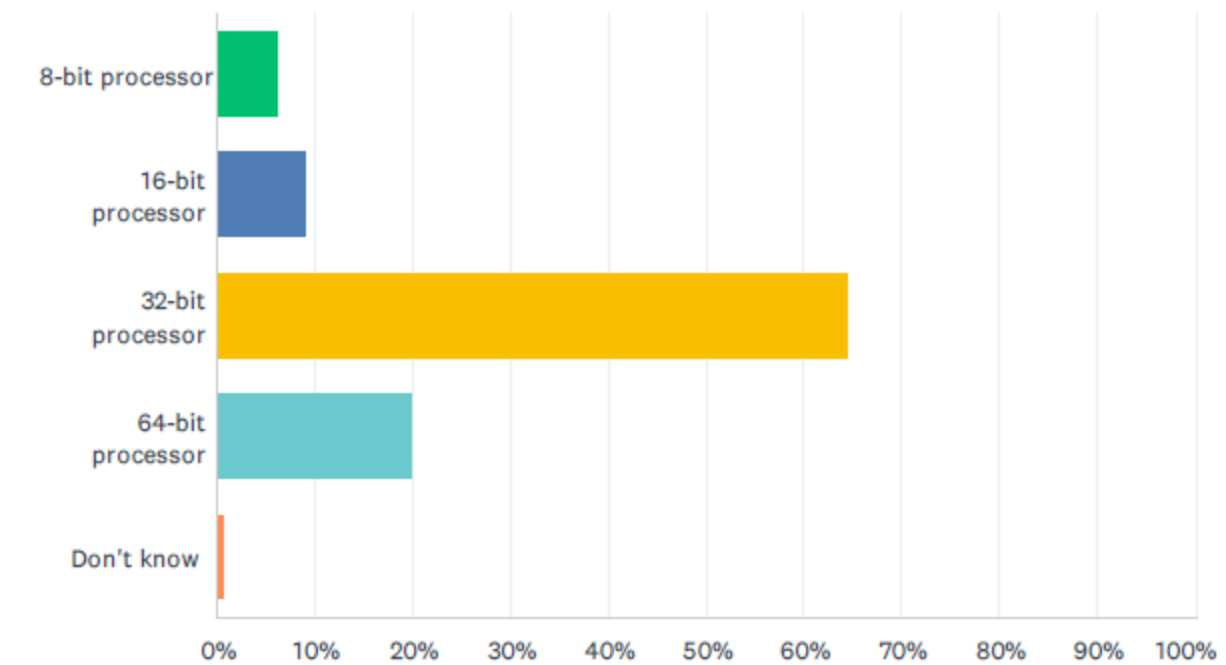
The survey shows that the majority of developers are not using embedded virtualization or hypervisors and do not plan to. However, for those who are or plan to, the adoption rate is growing. For those who are interested in virtualization and hypervisors, the primary drivers are separation of multiple applications and the need to support hard real-time applications and a guest operating system.



Processors (Part 1)



- ✓ The majority of projects use a single processor.
- ✓ A quarter of projects use two processors.
- ✓ Projects with 3-5 processors are also common.



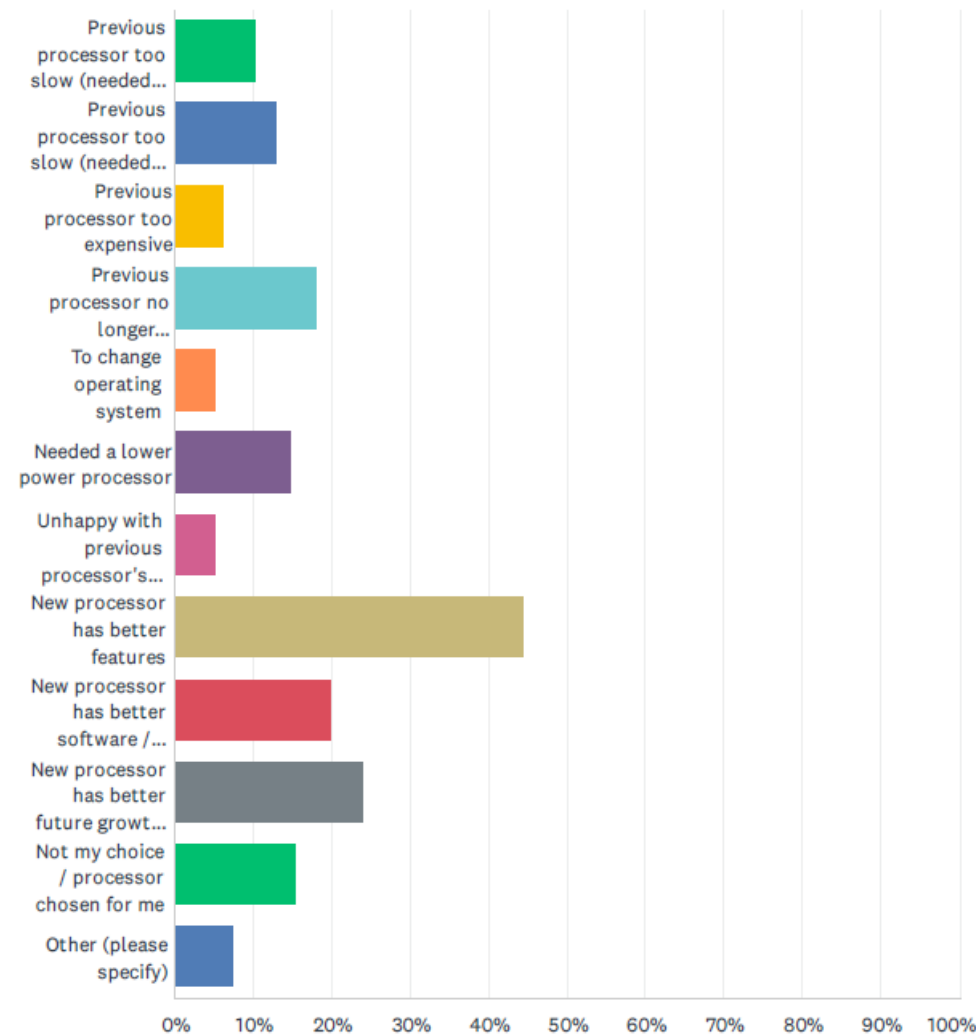
Moreover, the survey reveals that most projects do not have additional processors. However, for those that do, the additional processors are overwhelmingly **32-bit processors**.

- ✓ A quarter of projects do not have additional processors.
- ✓ Additional processors are primarily 32-bit.
- ✓ The use of 32-bit processors is consistent across both main and additional processors.
- ✓ Heterogeneous Multi-Chip Designs are a Top Choice
- ✓ Integrated Multicore SoCs are Widely Adopted
- ✓ Specialized AI and GPU Hardware are Niche

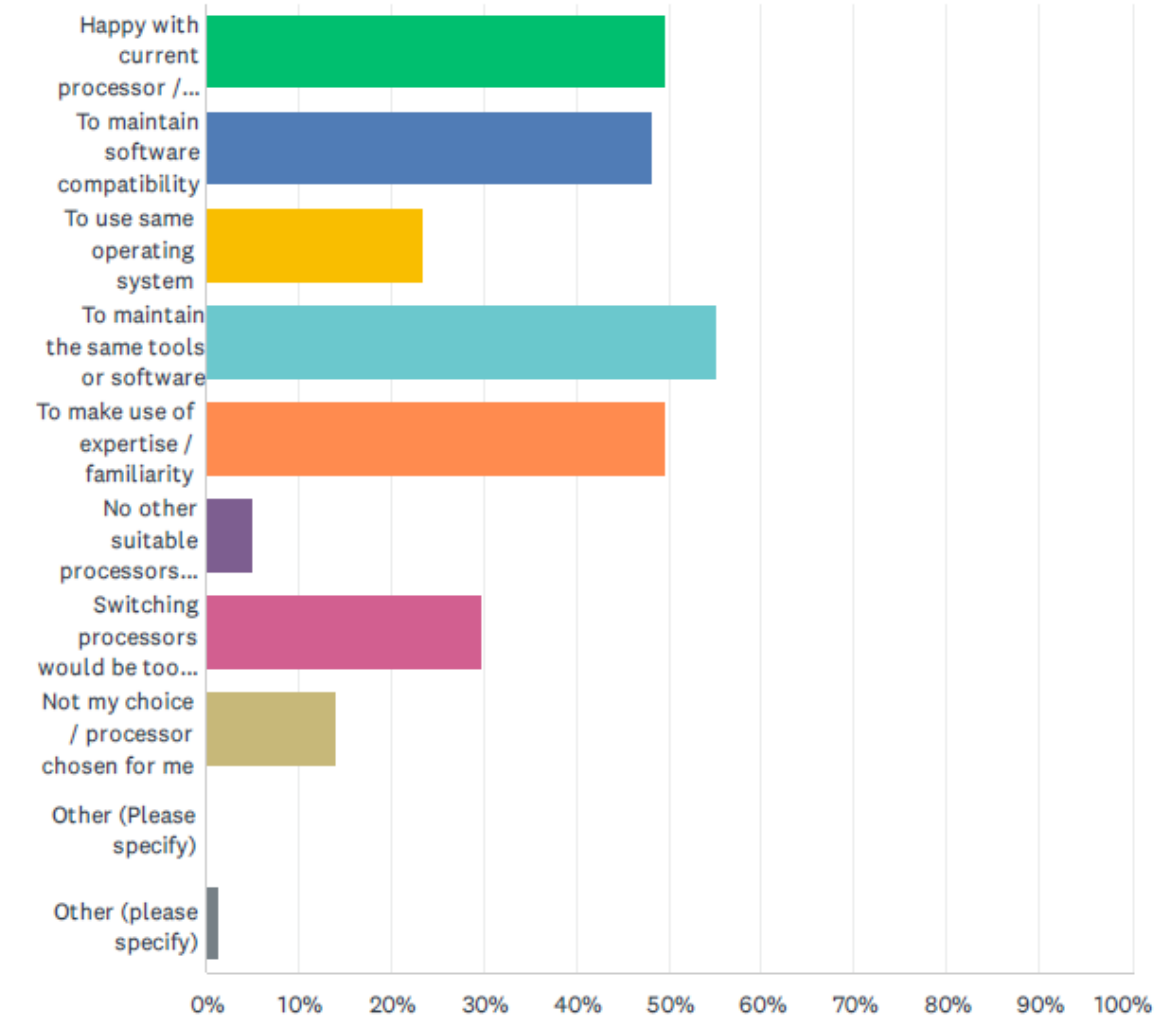
Processors (Part 2)

- ✓ The most common clock rate is 100 - 249 MHz.
- ✓ The second and third most common ranges are 250 - 499 MHz and 50 - 99 MHz.
- ✓ Most projects use processors in the low to mid-range of clock speeds.

The survey shows that a slight majority of developers, 58.25%, are reusing the same processor from a previous project. This practice is likely driven by the benefits of familiarity, cost savings, and the ability to reuse existing software and hardware designs. For those who reused a processor, the primary reasons are centered on efficiency and familiarity.



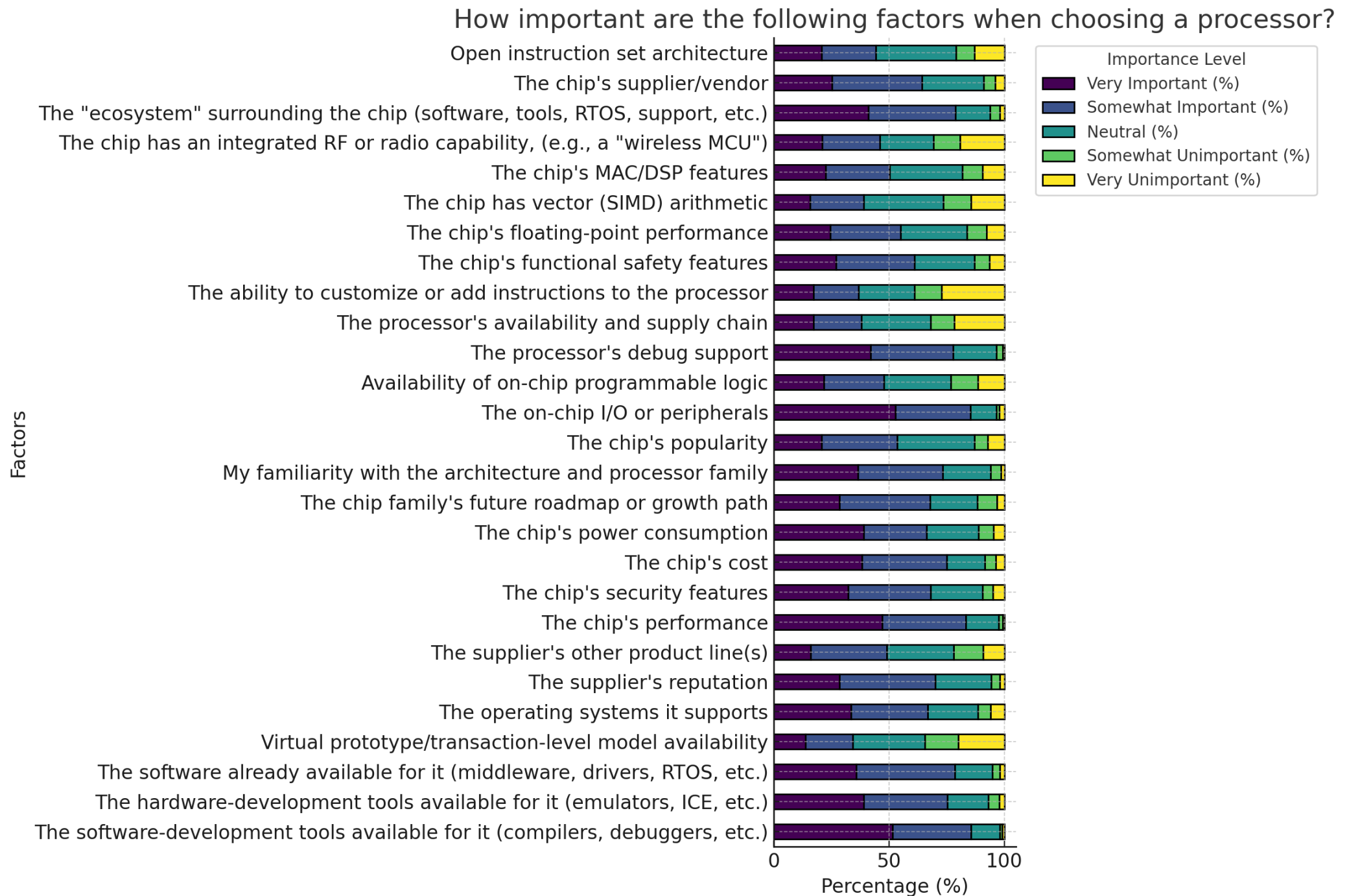
For the professionals who chose to switch processors, the main motivation was a desire for improved functionality and features.



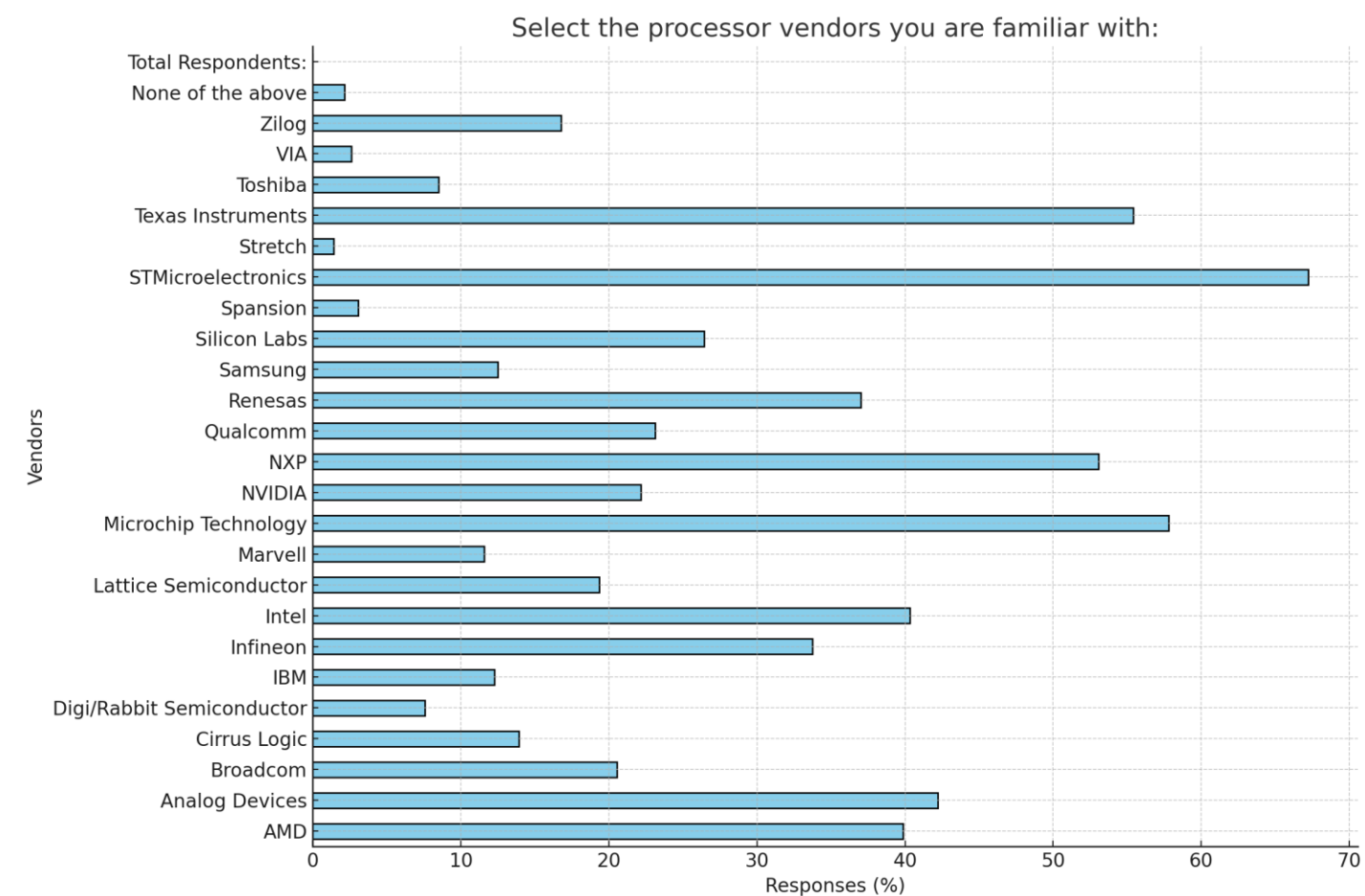
Processor Evaluation

When switching processors, the survey reveals that developers are almost evenly split between two paths. A slight majority chose a different processor from the same family, architecture, or instruction set. This suggests a preference for continuity and leveraging existing knowledge.

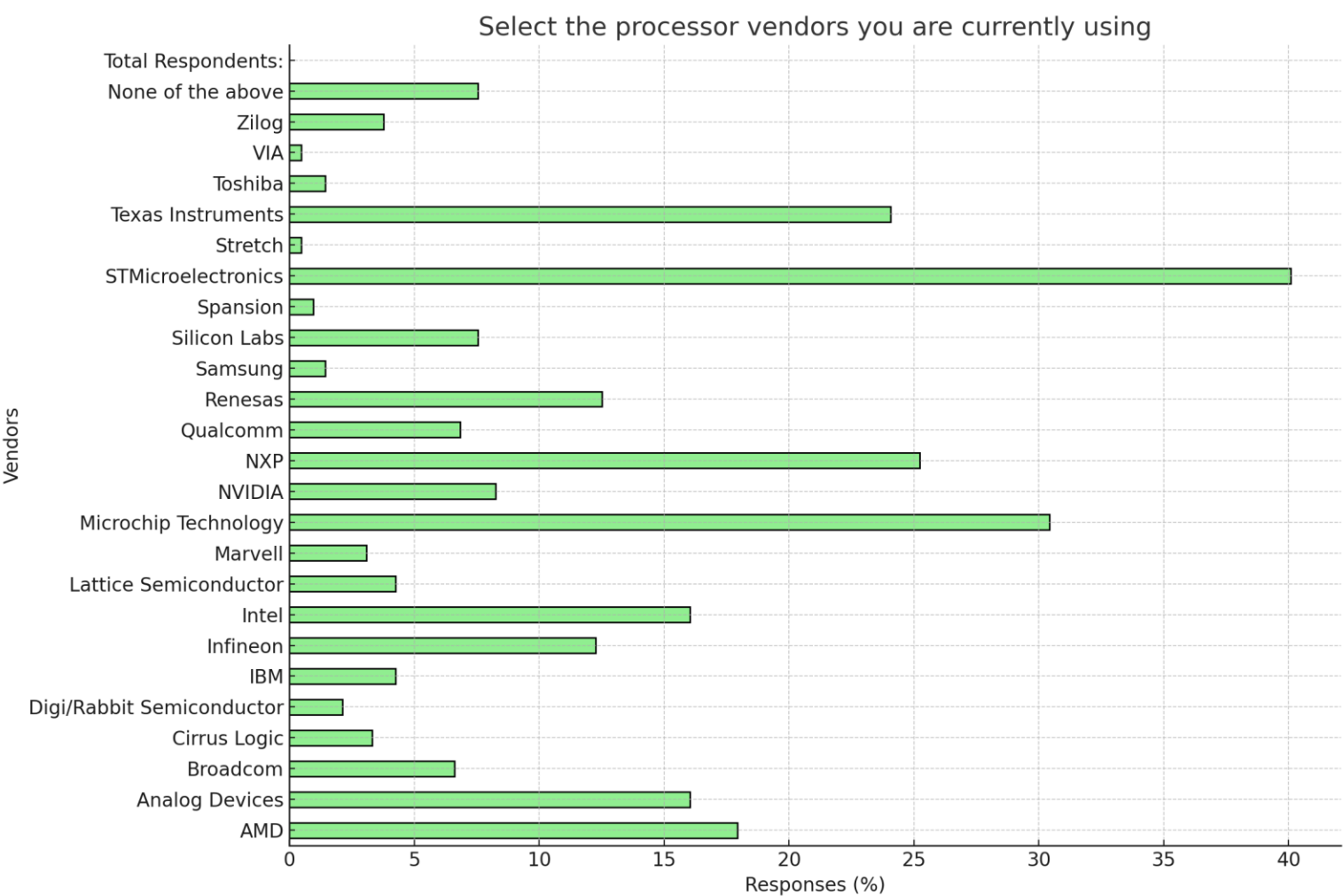
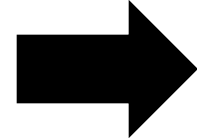
- ✓ A small majority of developers stay within the same processor family.
- ✓ Almost half of developers are willing to switch to a different architecture.
- ✓ This shows a balance between continuity and innovation.



Processor Vendors

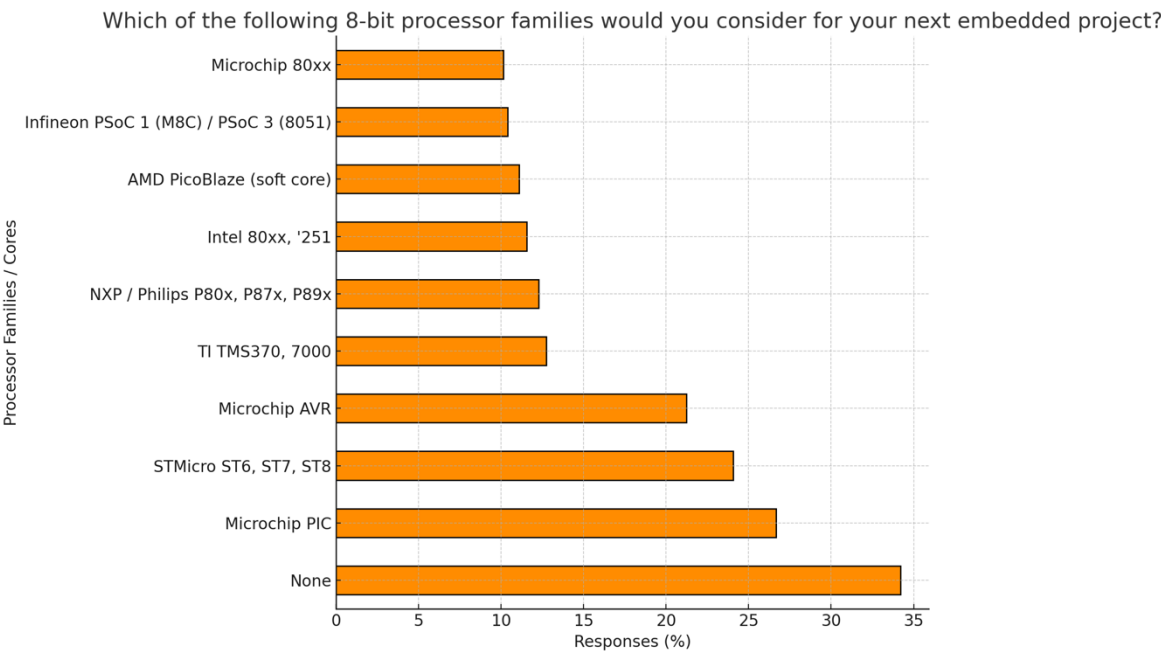


- ✓ Texas Instruments and Microchip are also top considerations.
- ✓ The data shows strong brand loyalty in the industry.

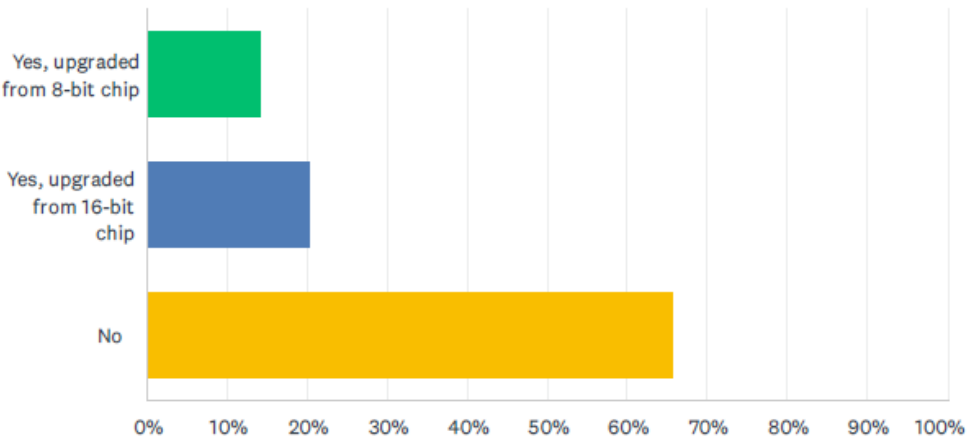
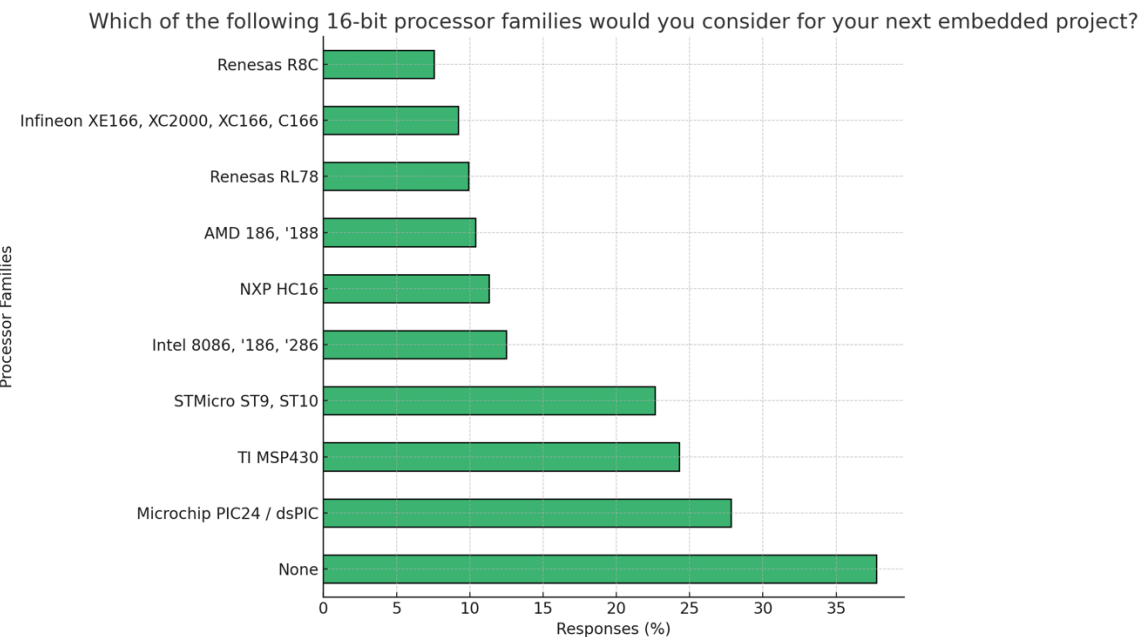
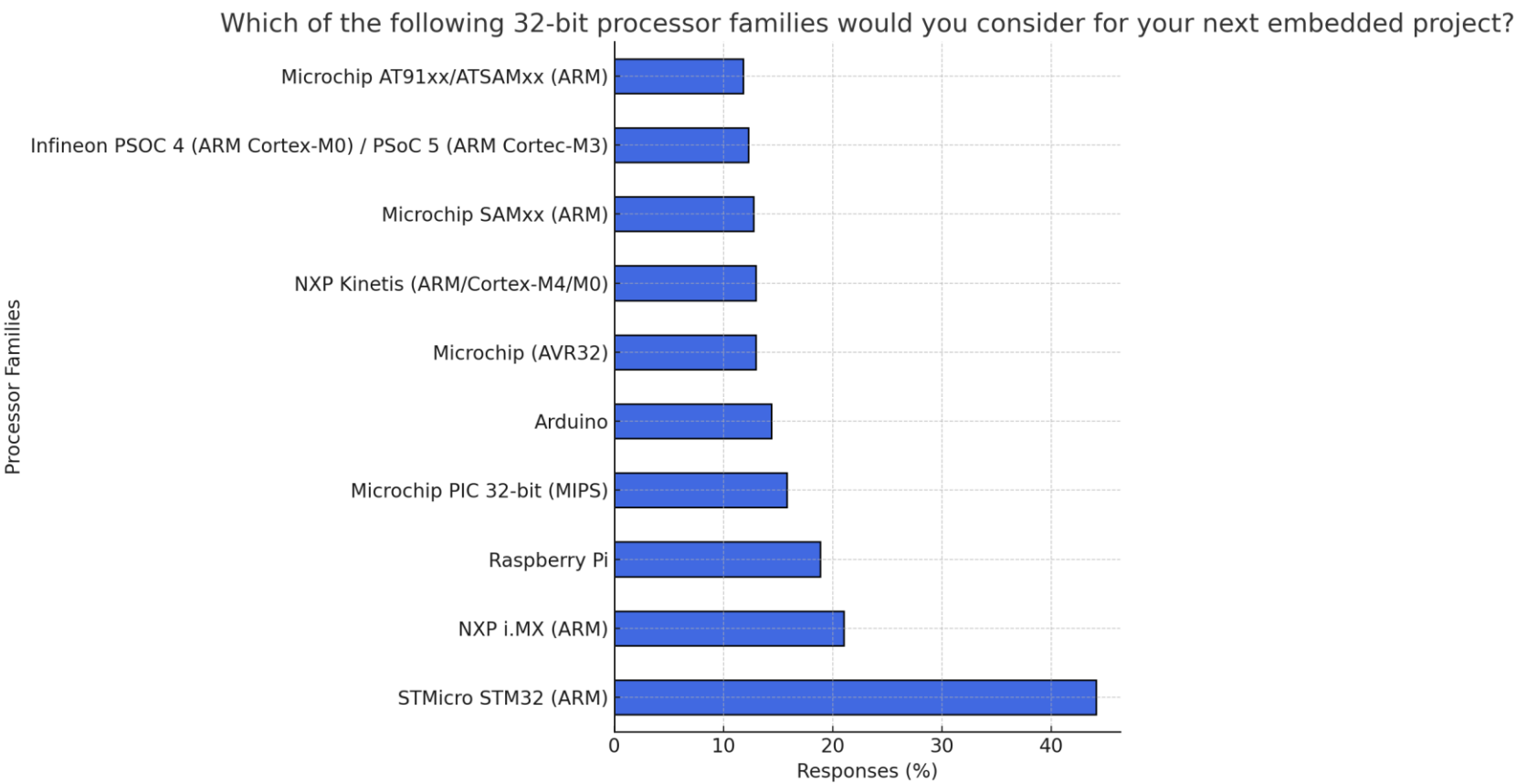


Specific Processor Families

Top 3 Most Considered 32-bit Processors: STMicro STM32 (ARM): 44.10% ; NXP i.MX (ARM): 20.99% ; Raspberry Pi: 18.87%



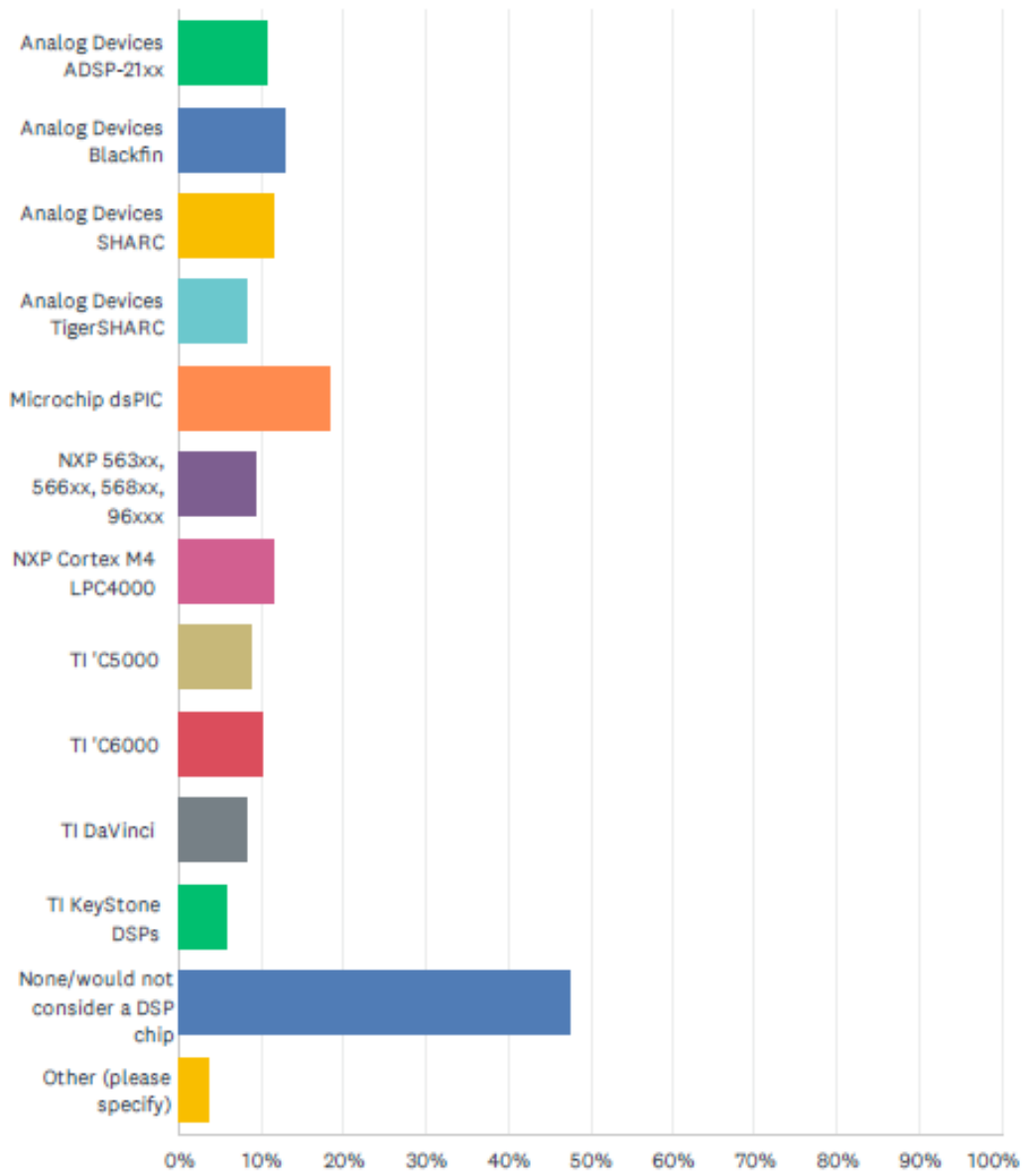
The survey data on 8-bit processor families reveals a significant trend away from these architectures.. This highlights the industry's shift towards more powerful 32-bit architectures to handle increasing project complexity. Similar to the 8-bit segment, a large percentage of respondents, over a third, would not consider using a 16-bit processor at all.



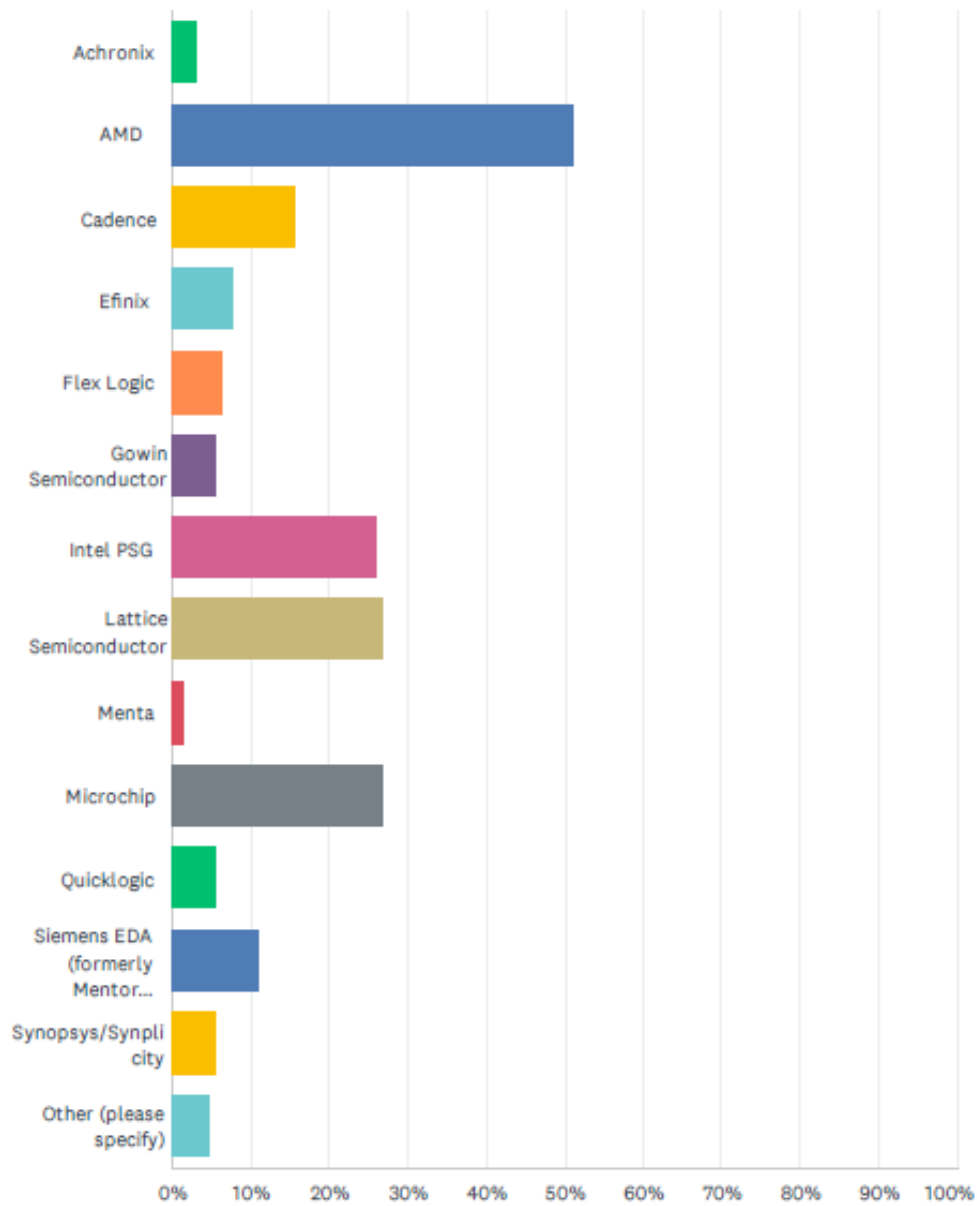
- ✓ The majority of professionals have not recently upgraded to a 32-bit chip.
- ✓ A significant portion has upgraded from a 16-bit chip.
- ✓ Fewer have upgraded from an 8-bit chip.

DSP & FPGA

The data shows that a significant portion of professionals, nearly half, would not consider a DSP chip at all for their next project.



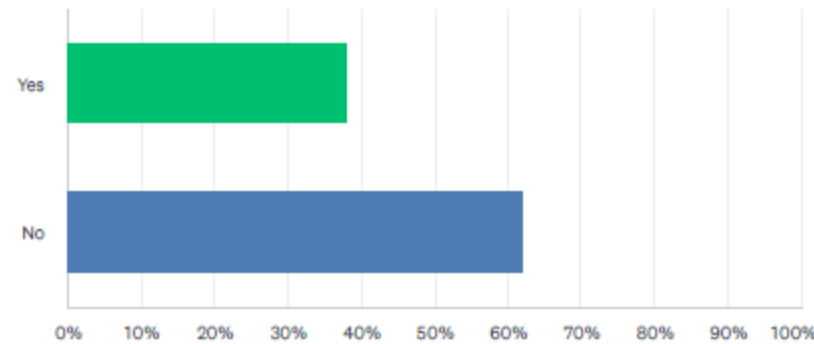
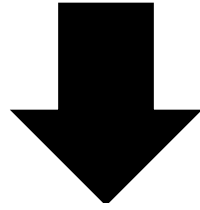
- ✓ The majority of embedded projects do not use an FPGA.
- ✓ FPGAs are used in a significant minority of projects.
- ✓ FPGAs are not yet a mainstream component in embedded design.



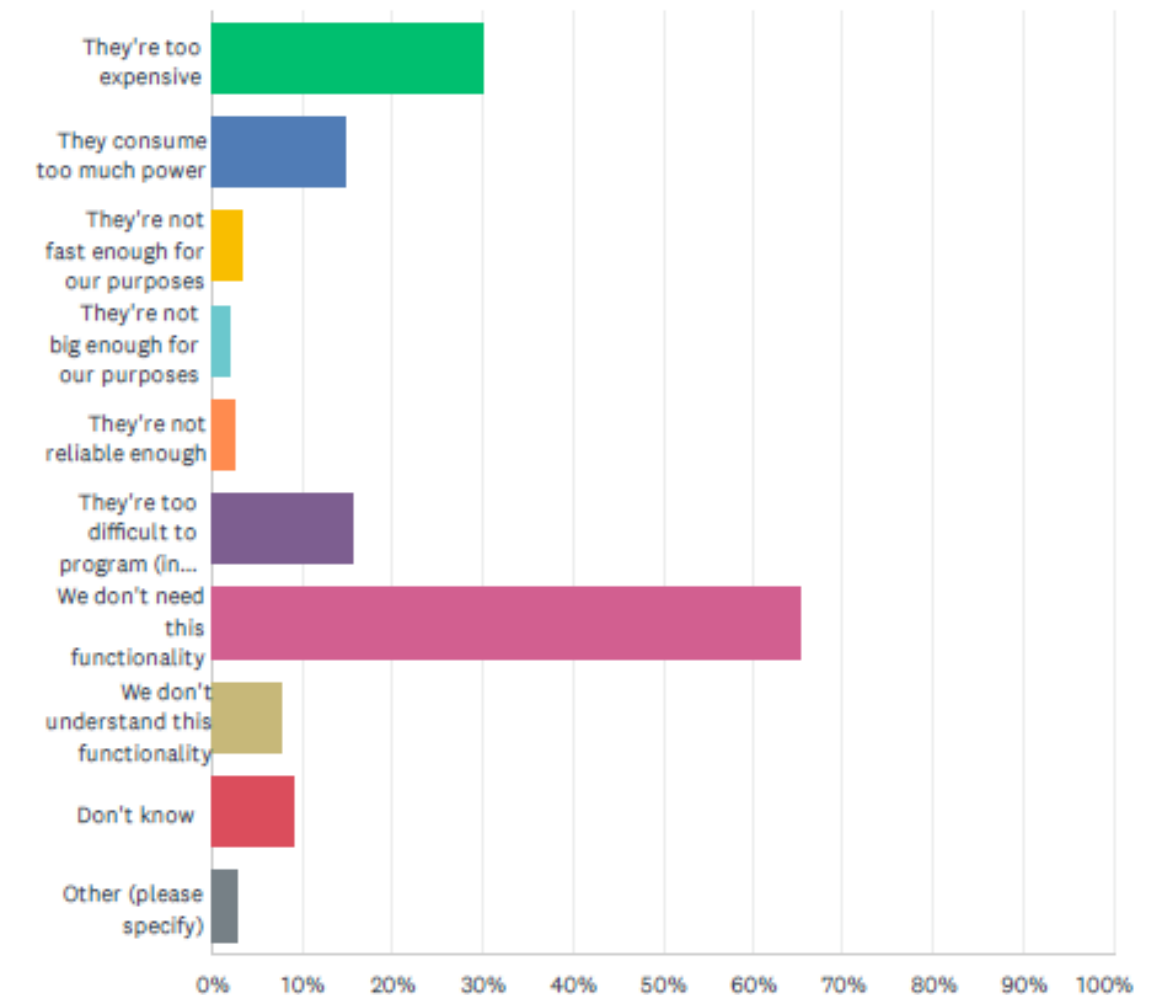
Among those who use FPGAs, AMD is the most used vendor. This is followed by Intel PSG and Lattice Semiconductor. The data suggests that these three companies are the primary players in the embedded FPGA market.

FPGA Use & Design Techniques

Will your next embedded project use an FPGA or programmable logic device?

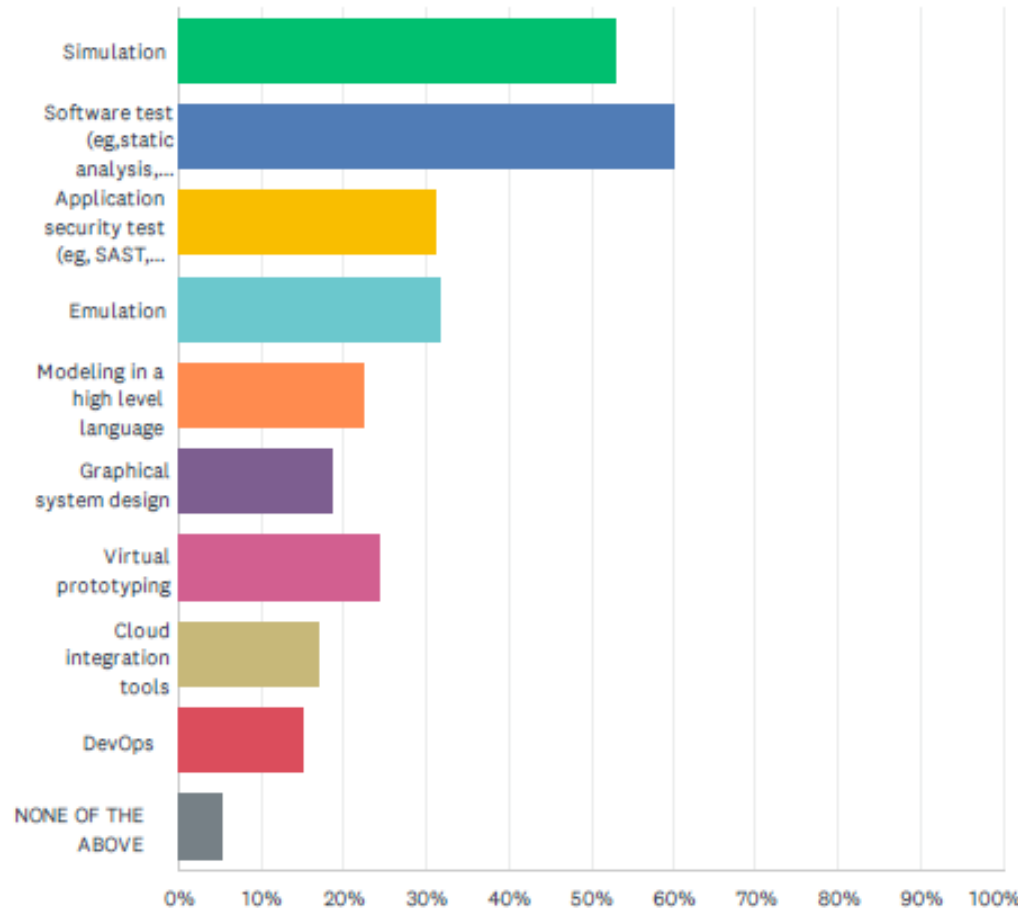


If no, why will your next embedded project NOT use an FPGA or programmable logic device?



Top 3 Most Important Future Design Techniques:

- Software test
- Simulation
- Emulation

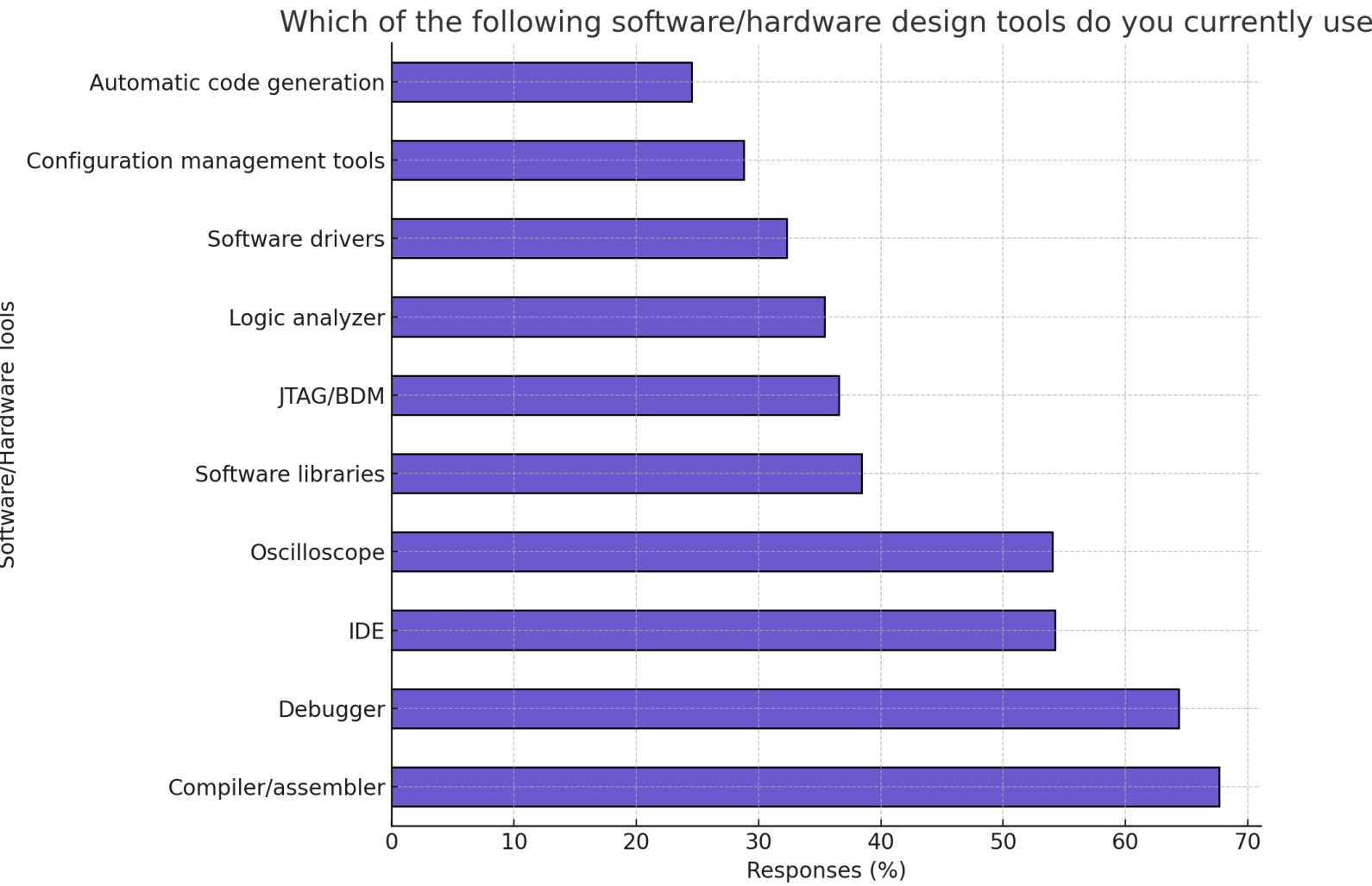


- ✓ The top reason for not using an FPGA is that the functionality is not needed.
- ✓ Cost is a major deterrent.
- ✓ The difficulty of programming FPGAs is also a significant barrier.

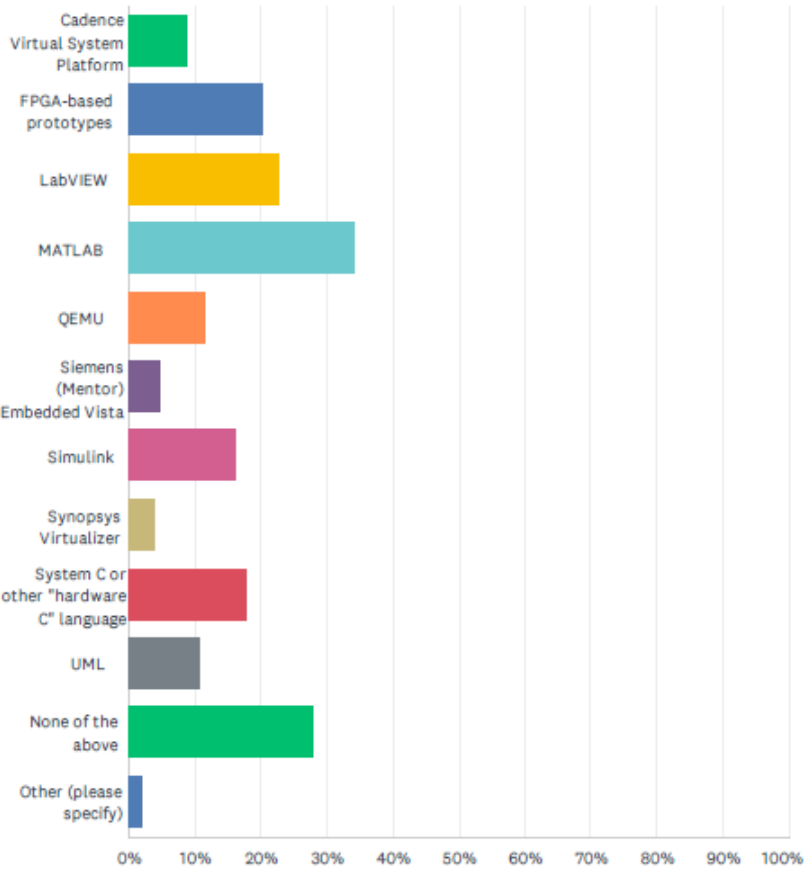
Tools, Professional Development

Software & Cloud Tools

- ✓ Compiler/assembler: 67.69%
- ✓ Debugger: 64.39%
- ✓ IDE: 54.25%

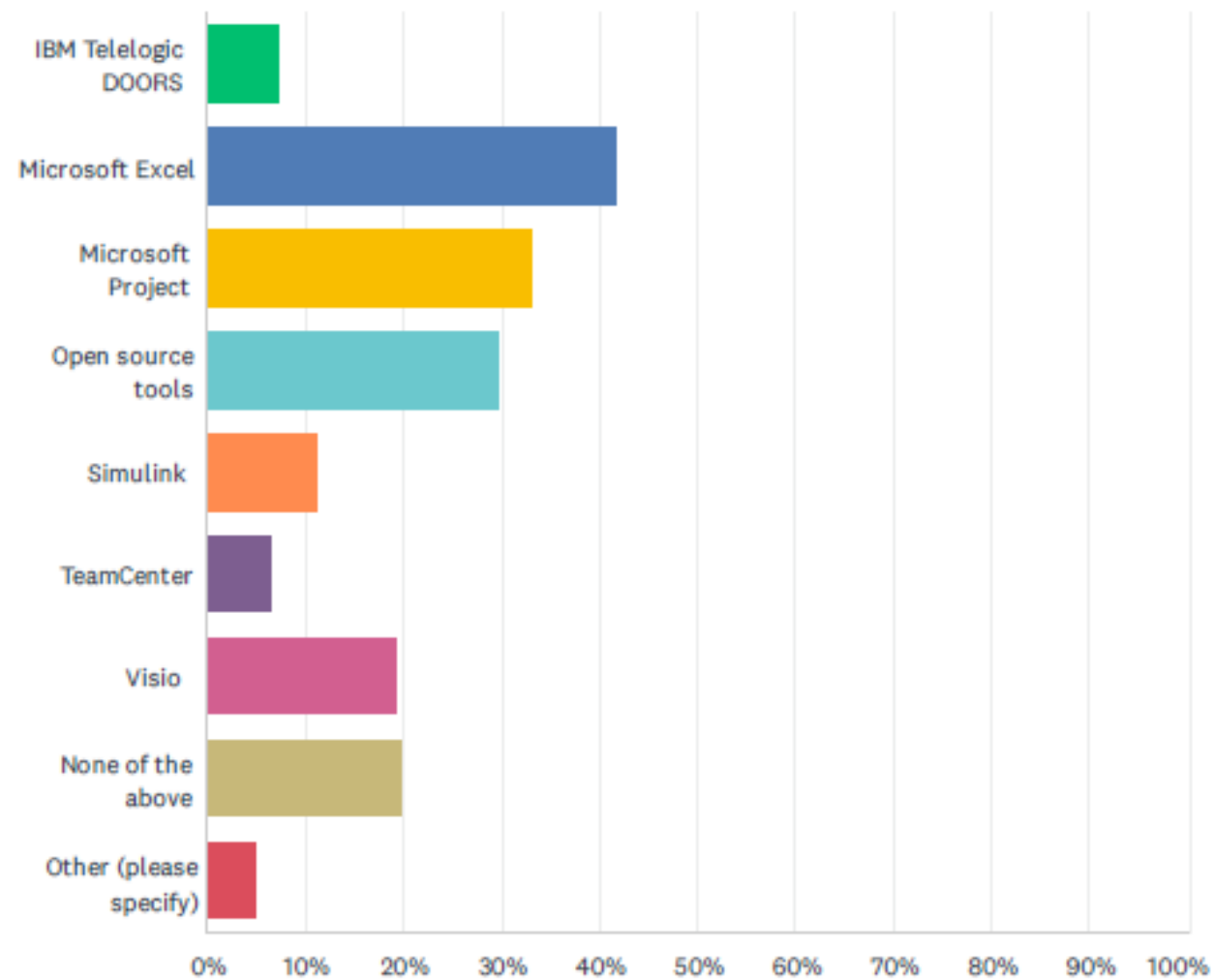


- ✓ MATLAB and LabVIEW are the most used system-level tools.
- ✓ A large number of organizations do not use the listed tools.
- ✓ This suggests a mix of commercial and in-house solutions.

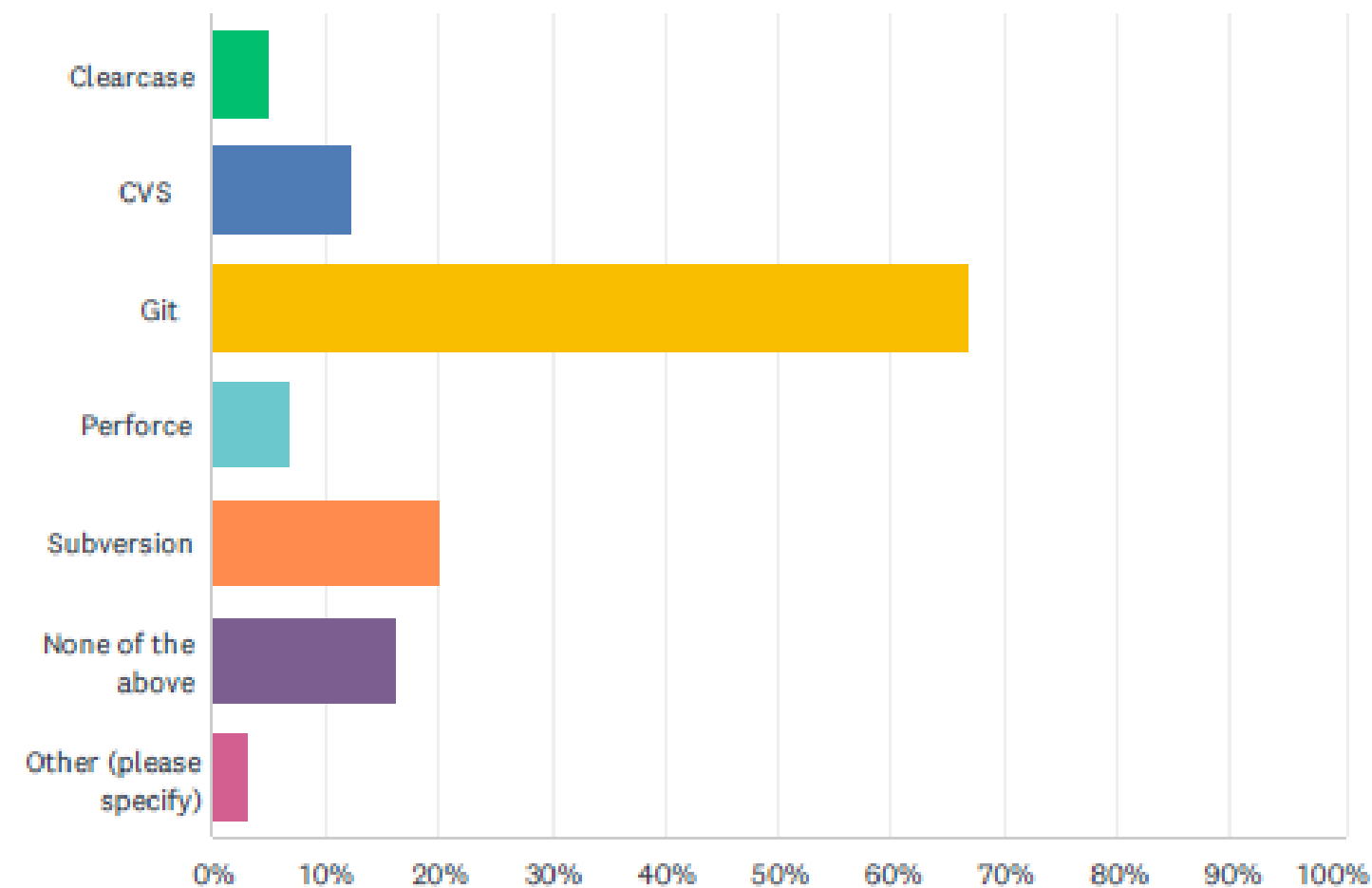


- ✓ Many organizations do not use cloud integration tools.
- ✓ Firmware updates are the most common use of these tools.
- ✓ Device and security management are also a major focus.

Management & Version Control

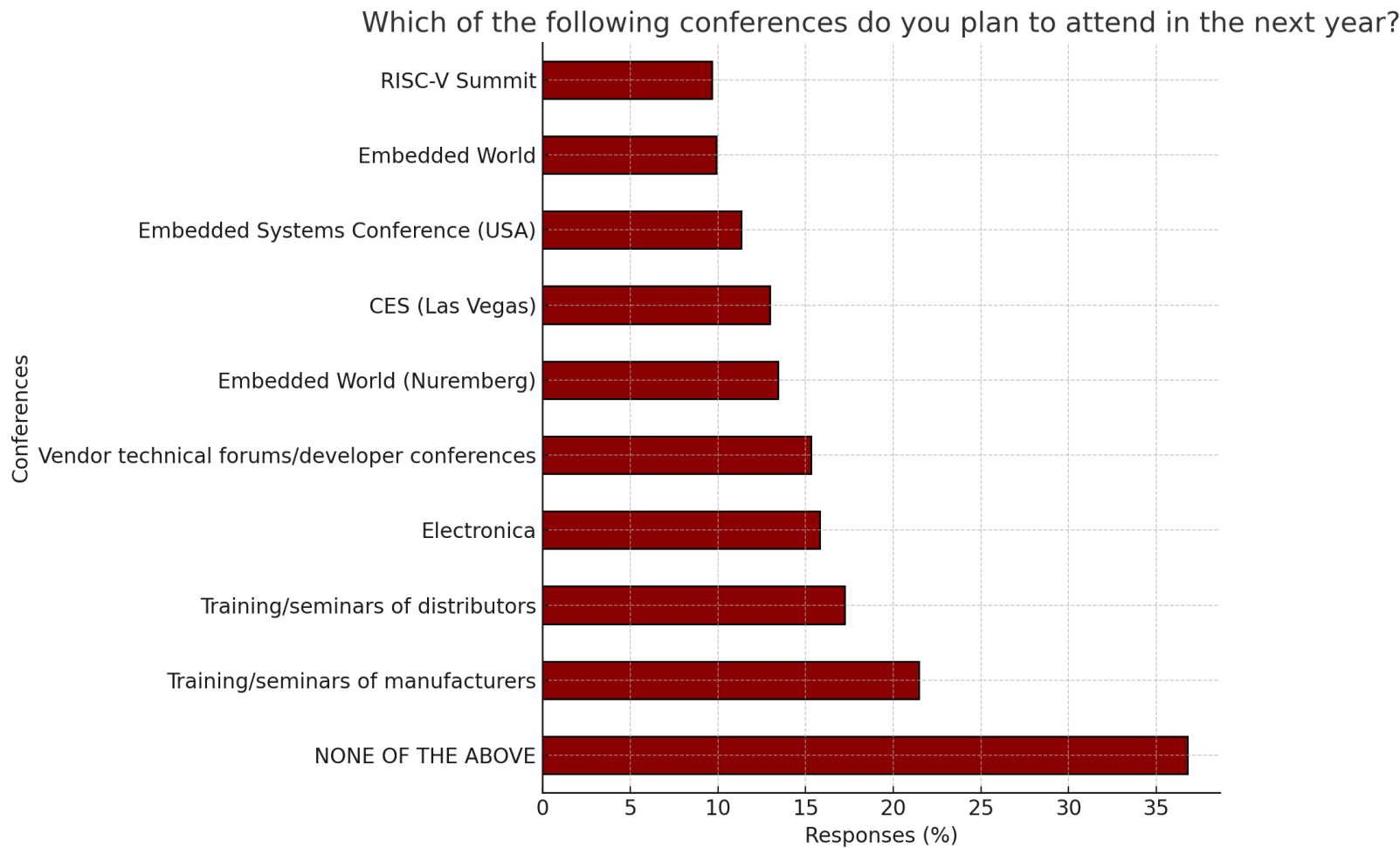


The survey reveals that professionals are using a variety of project management tools, with Microsoft Excel being the most common by a large margin. This is followed by Microsoft Project and open-source tools.



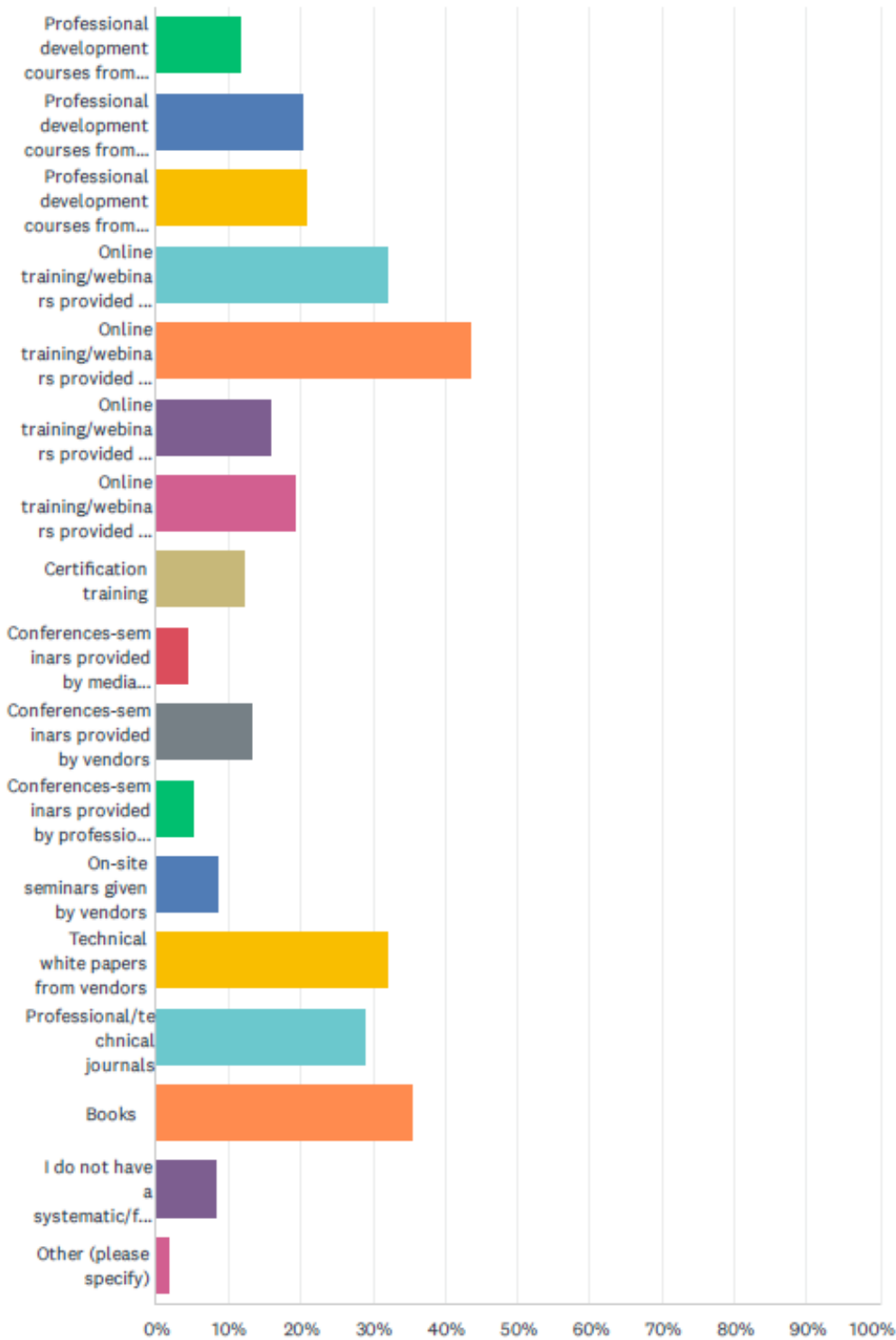
The survey data shows a clear and overwhelming preference for Git as a version control system.

Professional Development & Conferences

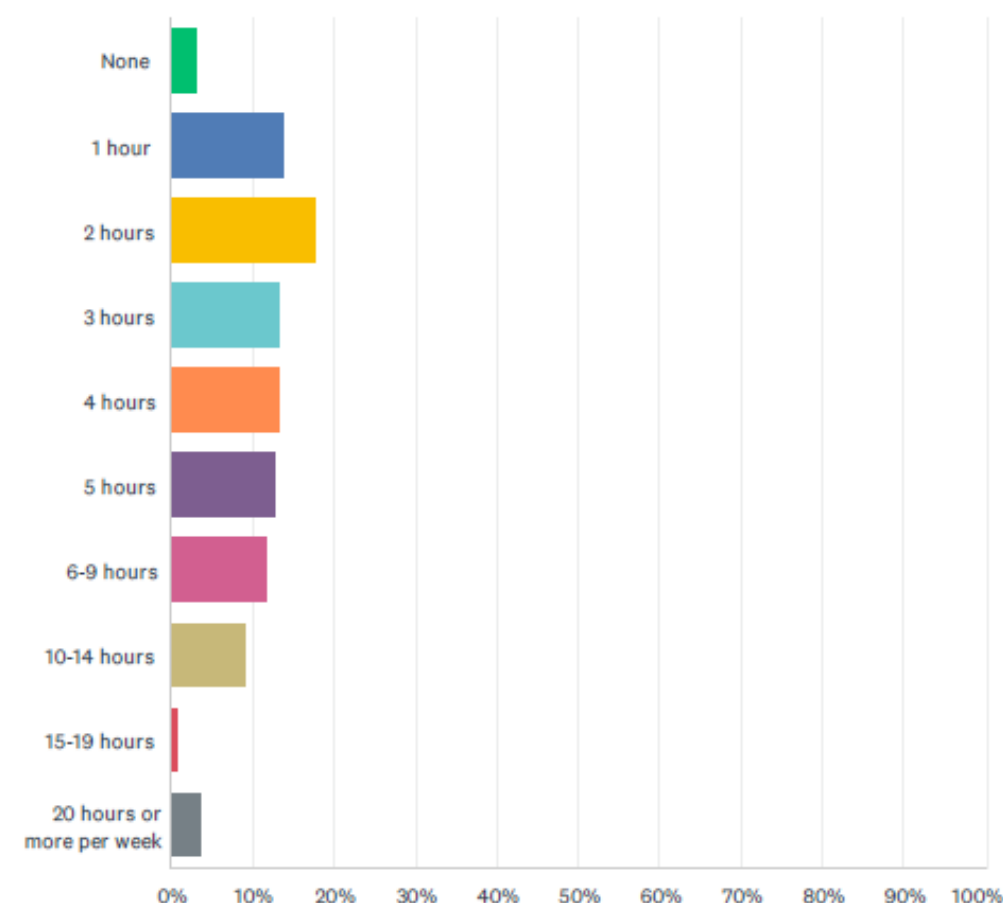


The survey shows that most embedded professionals dedicate a moderate amount of time to formal training. The most common response is 10-14 days per year, followed by 5 days and 6-9 days per year.

The survey provides insight into how professionals stay current in their field. The top three methods are online training/webinars provided by vendors, followed by books and technical white papers from vendors.

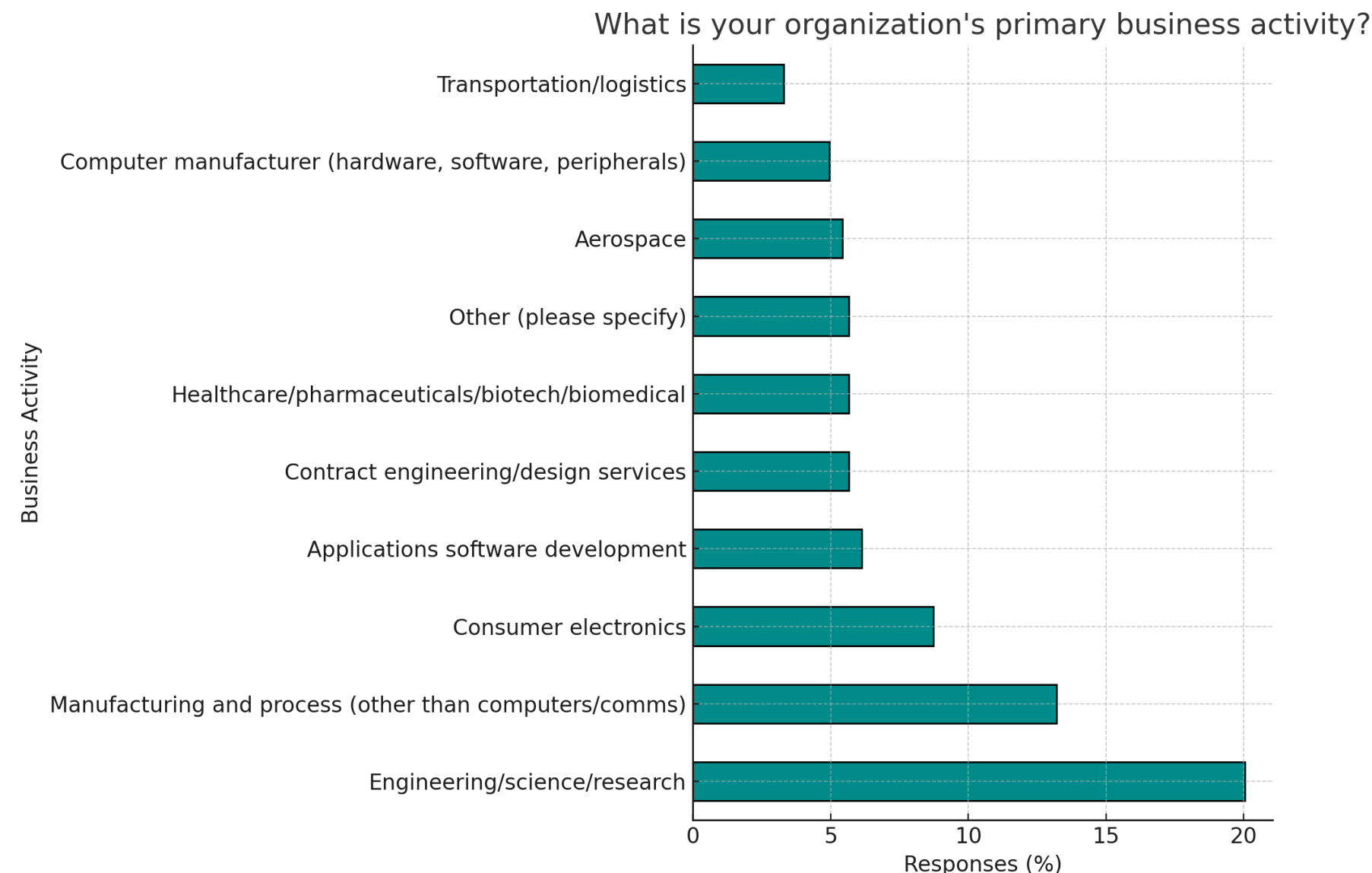


Reading & Organizational Info



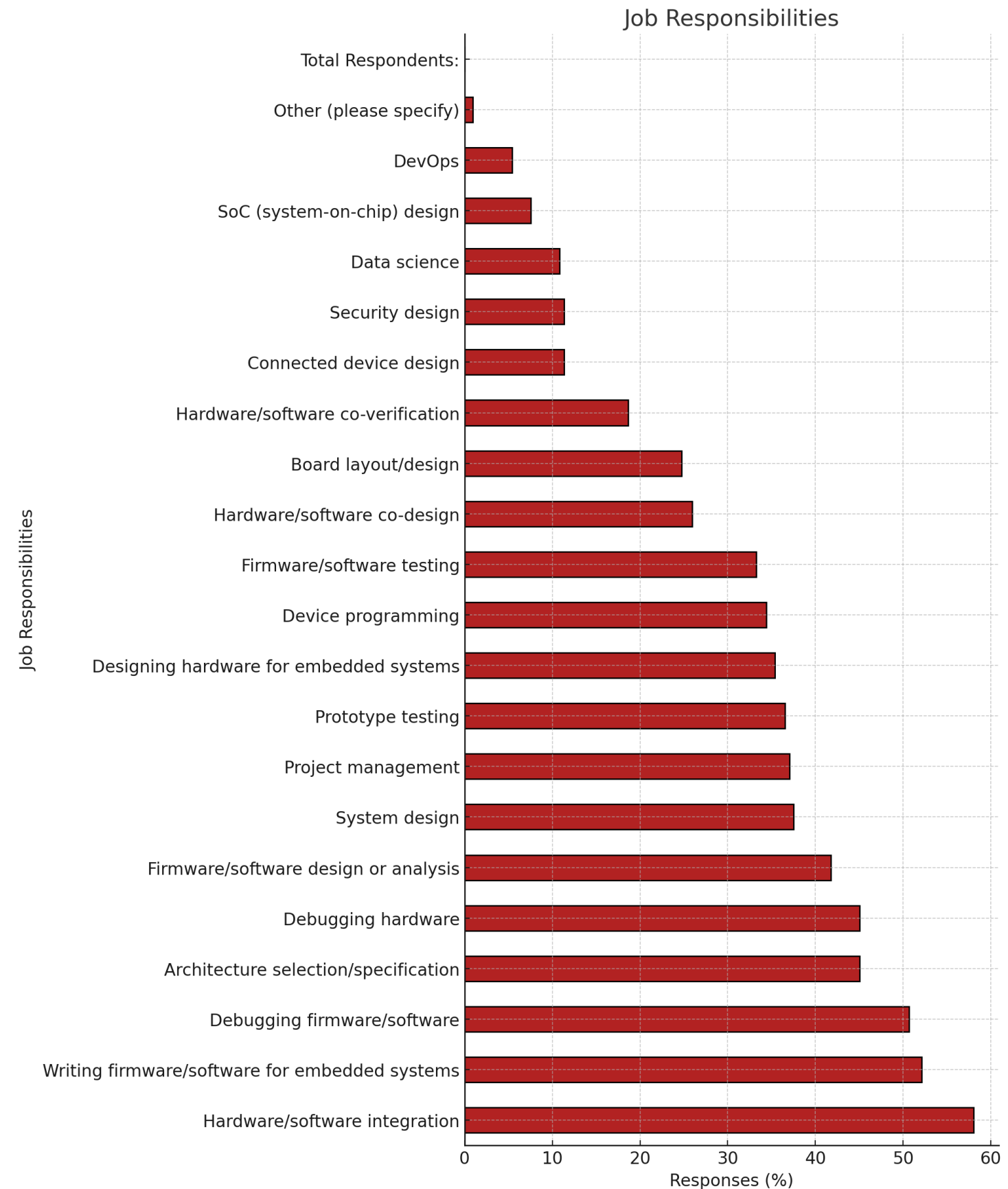
Most professionals read technical publications for 1-3 hours per week.

- ✓ The most common company size is 10-49 employees.
- ✓ Very small companies (1-9 employees) are also common.
- ✓ The industry is composed of a mix of company sizes.



Job Responsibilities

- ✓ Hardware/software integration is the top responsibility.
- ✓ Writing and debugging firmware/software are also key tasks.
- ✓ Job responsibilities are primarily technical and hands-on.



Observations

Observations

- ✓ **Most projects are upgrades, not new designs** – The majority of work focuses on improving existing solutions.
- ✓ **Teams dominated by hardware & firmware engineers** – Core expertise lies in technical development roles.
- ✓ **Top challenges: security, debugging, performance** – Critical hurdles that slow down design cycles.
- ✓ **Strong reliance on open-source OS and code reuse** – Cost efficiency and flexibility drive adoption.
- ✓ **Growing interest in AI, ML, and advanced features** – Emerging technologies gaining momentum for future designs.

Conclusion

Conclusion

- ✓ **Industry balancing maturity with innovation** – Building on proven practices while adopting new technologies.
- ✓ **Embedded systems have global, cross-industry relevance** – Used in industrial, IoT, automotive, and beyond.
- ✓ **Future shaped by efficiency, security, intelligence** – Key factors guiding the next generation of solutions.

Thank You

EMBEDDED.COM