embedded survey✓

The Current State of Embedded Development

MAY 2023

Methodology

Online survey

Americas	53%
North America	49
South America	3
Central America	1

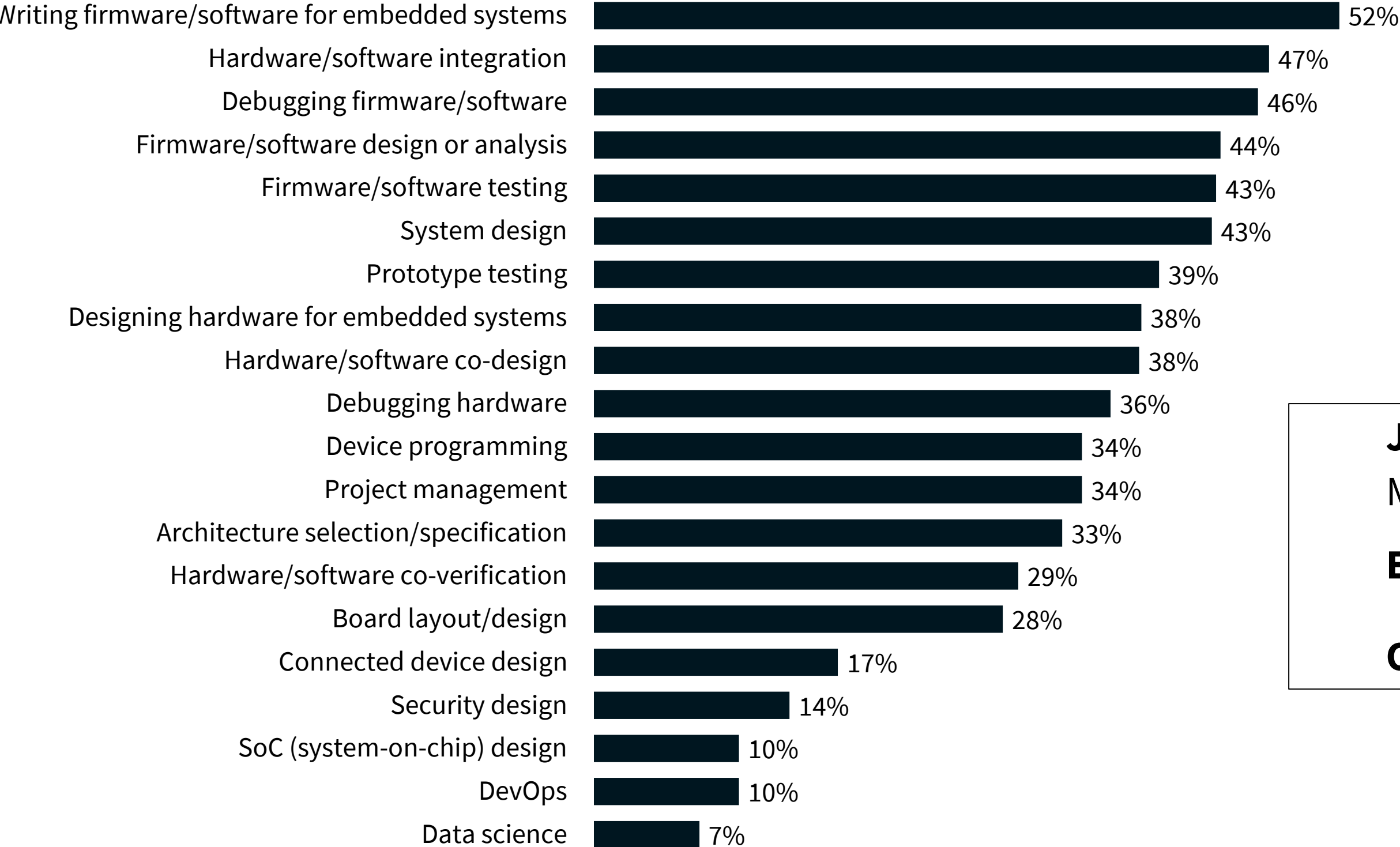| EMEA	23%
Germany	3
Italy	3
UK	3
France	2
Other EMEA	12

APAC	24%
India	8
Taiwan	7
China	4
Other APAC	5

- **Field Dates**: Feb 9 to March 3, 2023

- Respondents screened for **engineering** responsibilities and **experience with embedded applications**

- Results based on **655 responses** (confidence level +/- 3.7%)

Total Respondents

Respondent Characteristics

Job responsibilities, experience and organization size

Job responsibility	%
Writing firmware/software for embedded systems	52%
Hardware/software integration	47%
Debugging firmware/software	46%
Firmware/software design or analysis	44%
Firmware/software testing	43%
System design	43%
Prototype testing	39%
Designing hardware for embedded systems	38%
Hardware/software co-design	38%
Debugging hardware	36%
Device programming	34%
Project management	34%
Architecture selection/specification	33%
Hardware/software co-verification	29%
Board layout/design	28%
Connected device design	17%
Security design	14%
SoC (system-on-chip) design	10%
DevOps	10%
Data science	7%

Job function: Staff (58%), Management (41%), Other (1%)

Experience: 18.9 years

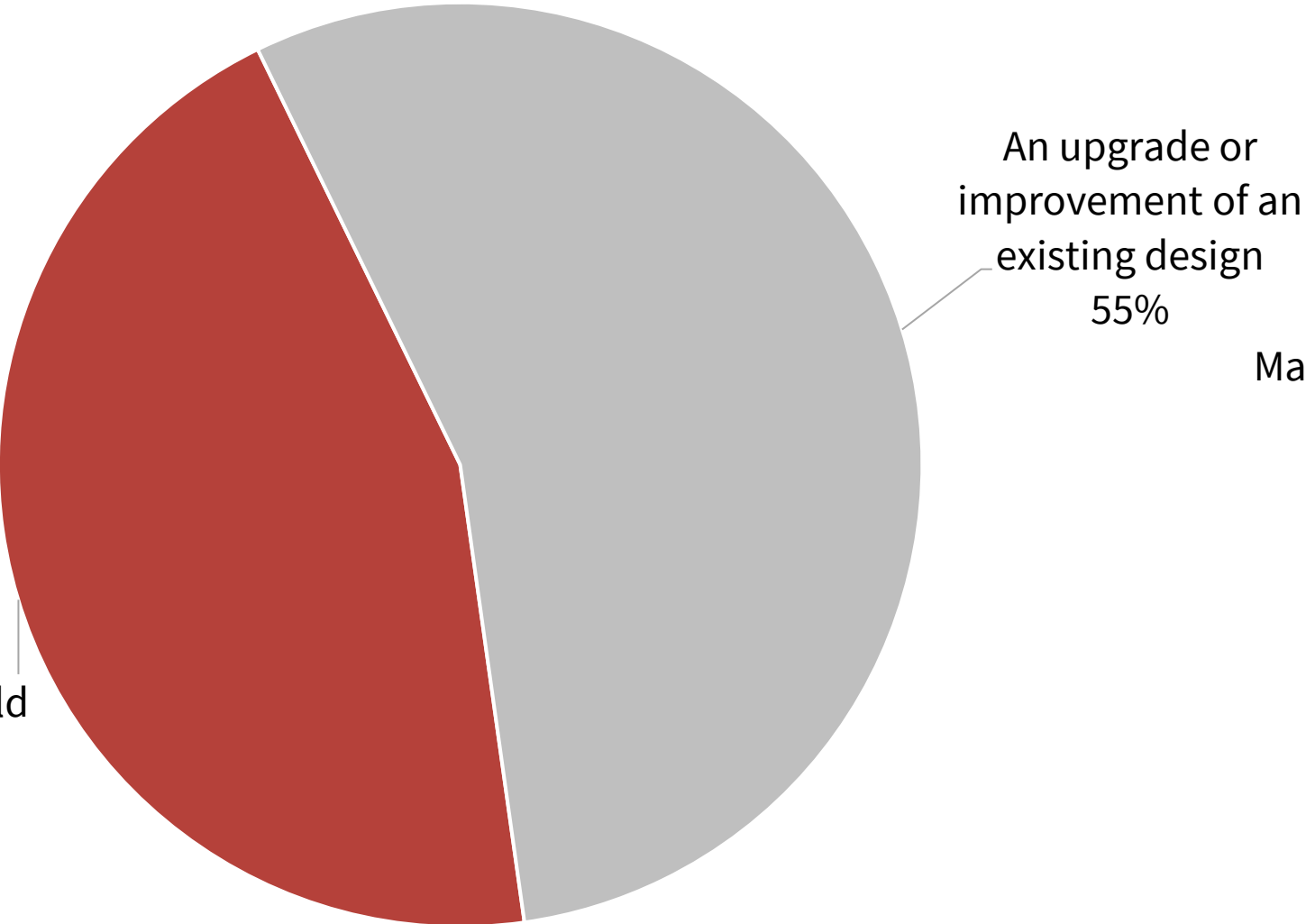**Organization size:** 3,288 employees

Total Respondents

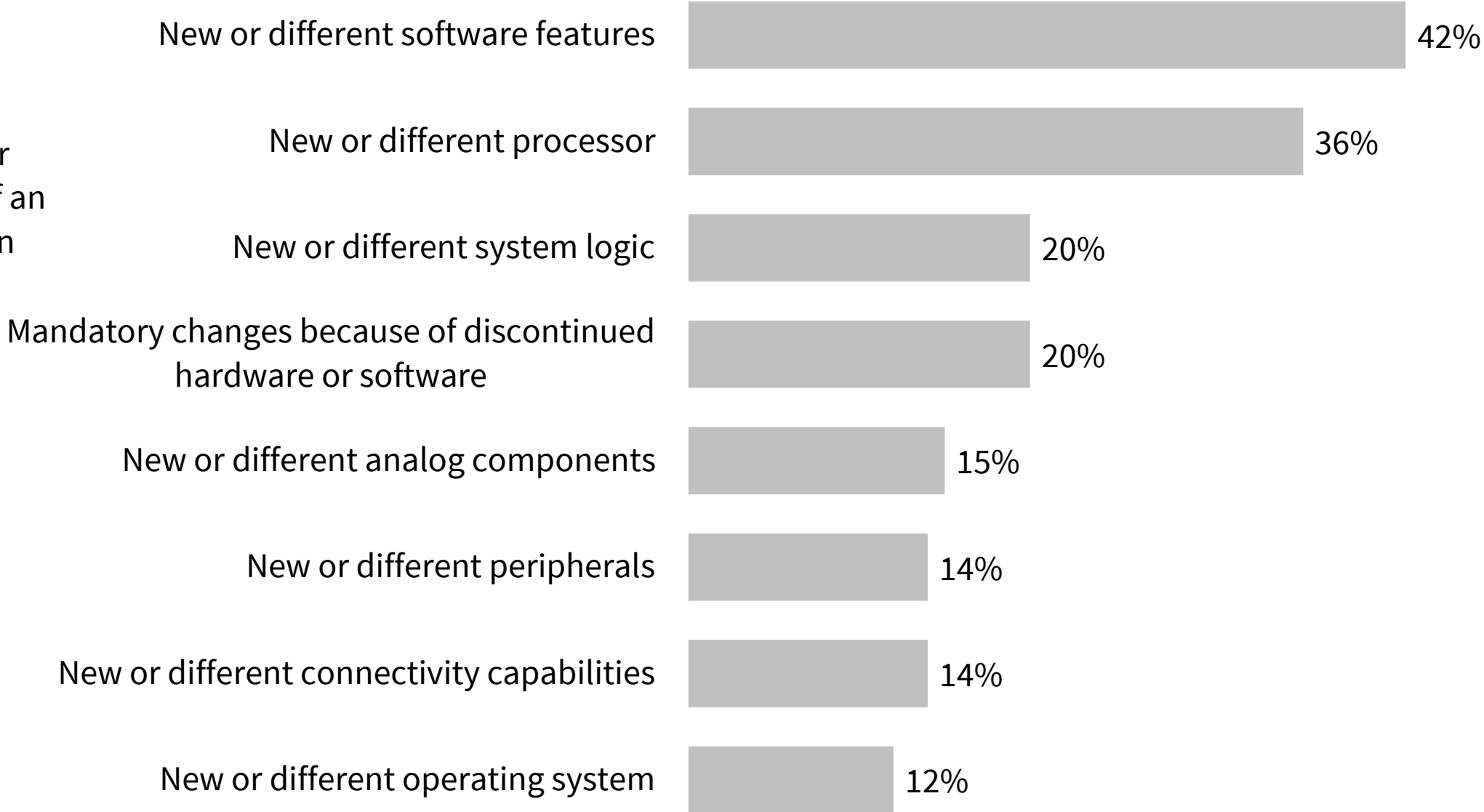# The Embedded Development Environment

Most embedded projects involve incremental upgrades to existing designs

Improvements including additional software features and/or better MPUs/MCUs (particularly by larger OEMs)
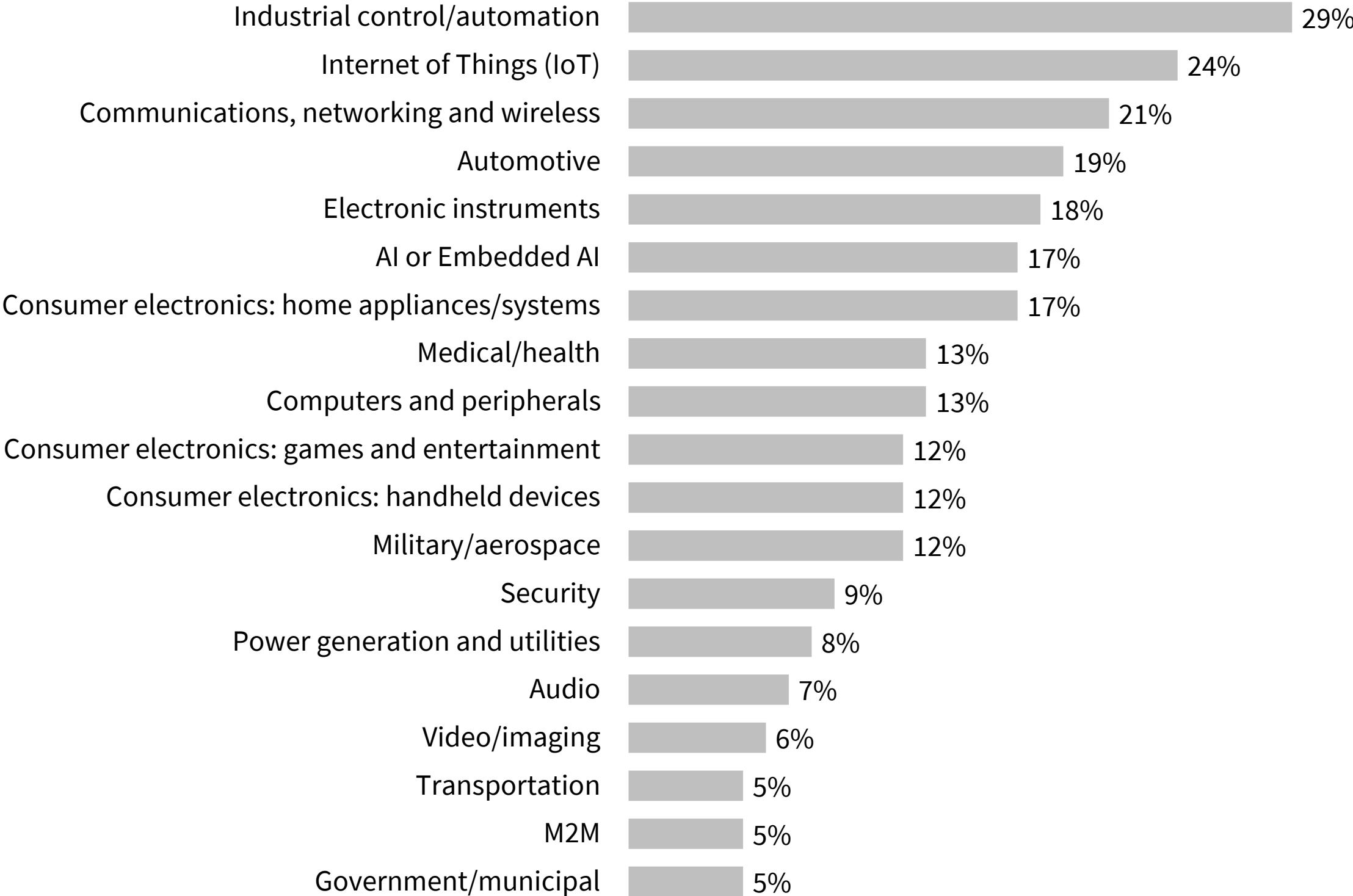
Current Embedded Project Is…

An upgrade or improvement of an existing design
55%

New to the world
45%

Total Respondents

Improvements for Upgrade

New or different software features	42%
New or different processor	36%
New or different system logic	20%
Mandatory changes because of discontinued hardware or software	20%
New or different analog components	15%
New or different peripherals	14%
New or different connectivity capabilities	14%
New or different operating system	12%

Base = Those upgrading existing design (362)

1. My current embedded project is:
2. Which two improvements were the main reasons for the upgrade?

ASPENCORE | 5

Embedded projects target a wide range of applications

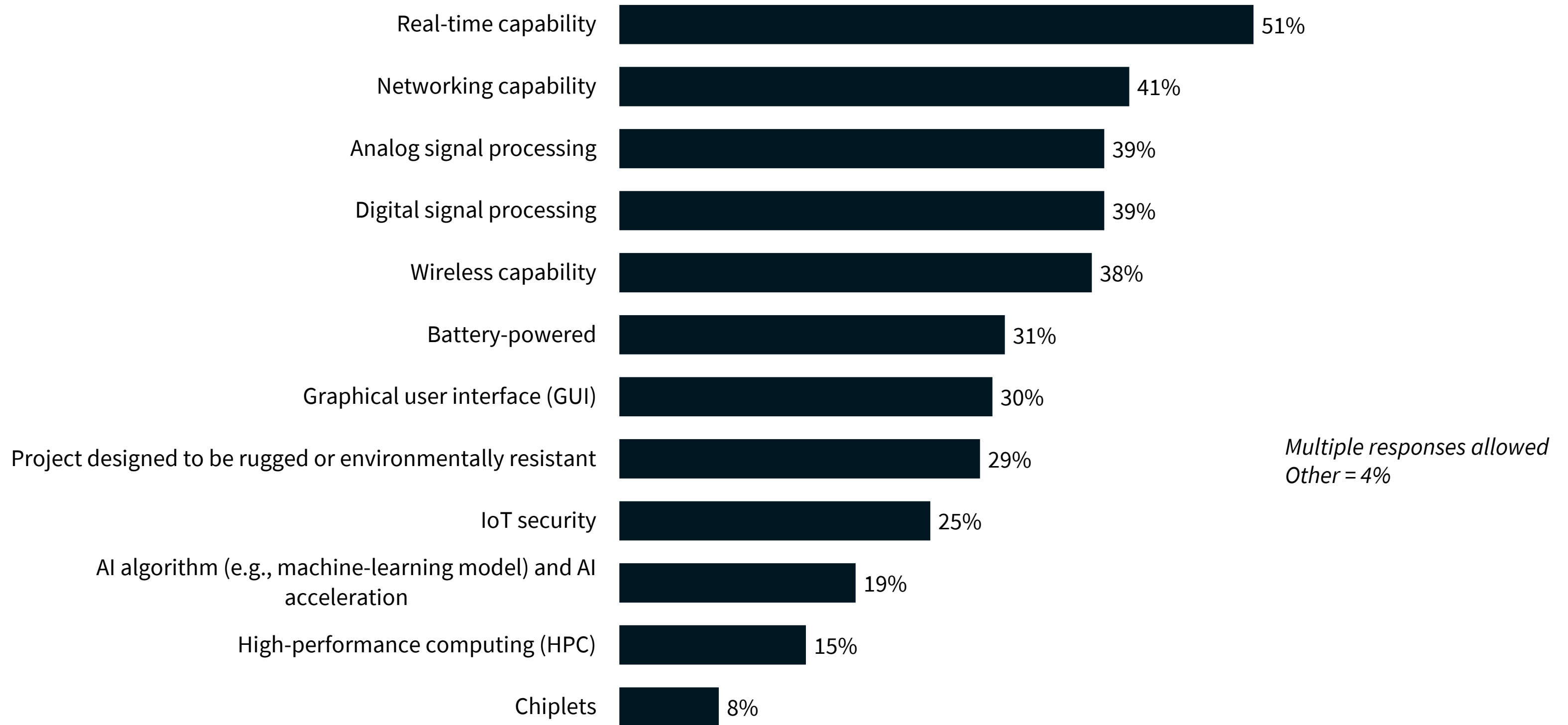Most projects are developed for *industrial automation and instrumentation, IoT, communications, and automotive*

Application	Percentage
Industrial control/automation	29%
Internet of Things (IoT)	24%
Communications, networking and wireless	21%
Automotive	19%
Electronic instruments	18%
AI or Embedded AI	17%
Consumer electronics: home appliances/systems	17%
Medical/health	13%
Computers and peripherals	13%
Consumer electronics: games and entertainment	12%
Consumer electronics: handheld devices	12%
Military/aerospace	12%
Security	9%
Power generation and utilities	8%
Audio	7%
Video/imaging	6%
Transportation	5%
M2M	5%
Government/municipal	5%

Multiple responses allowed

Total Respondents

embedded survey

6. For what types of applications are your embedded projects developed?

ΛSPENCORE | 6

Current embedded development devotes considerable attention to performance, connectivity, power efficiency and signal processing

EMEA and APAC teams are particularly interested in these capabilities
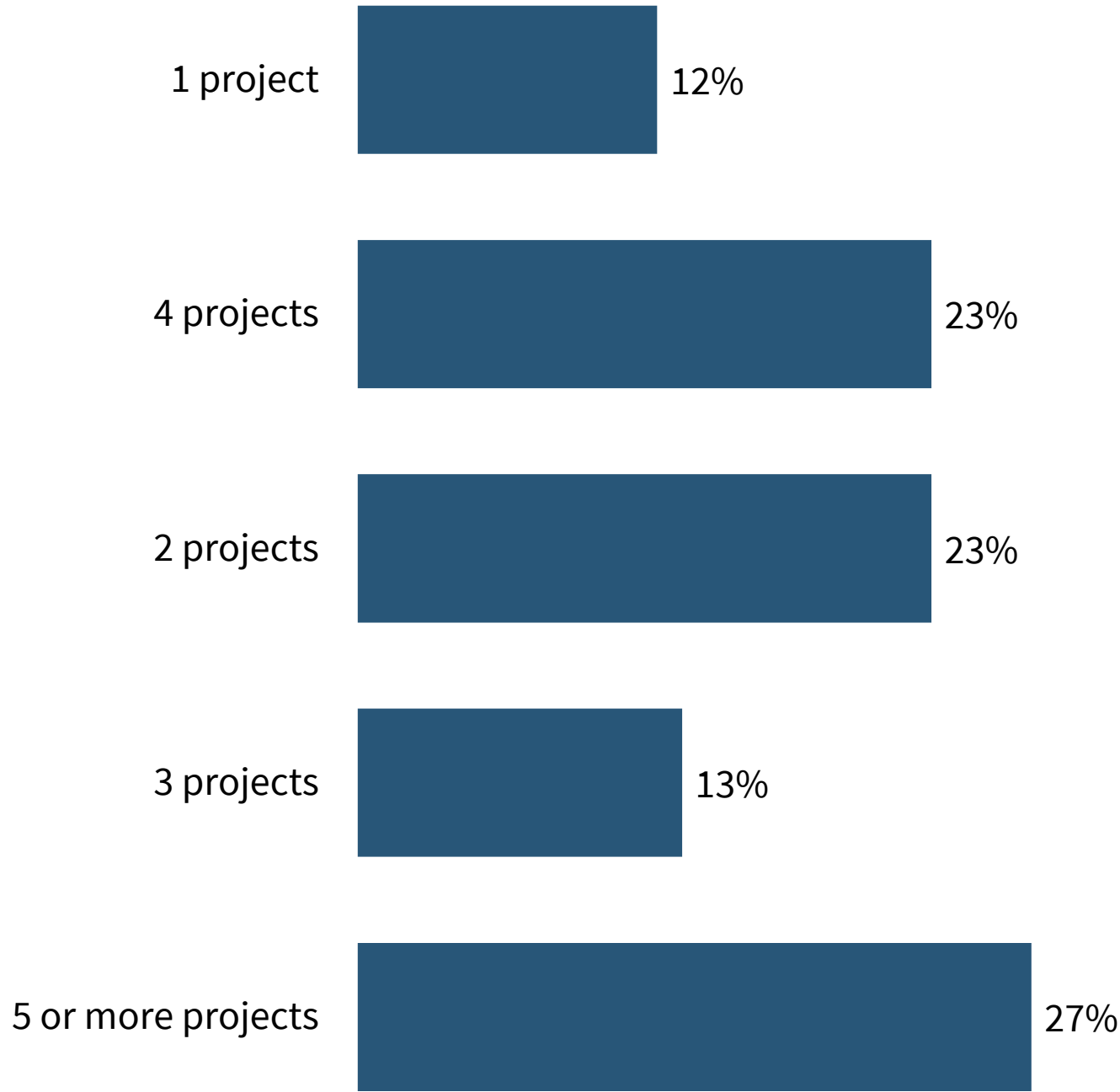
Capability	Percentage
Real-time capability	51%
Networking capability	41%
Analog signal processing	39%
Digital signal processing	39%
Wireless capability	38%
Battery-powered	31%
Graphical user interface (GUI)	30%
Project designed to be rugged or environmentally resistant	29%
IoT security	25%
AI algorithm (e.g., machine-learning model) and AI acceleration	19%
High-performance computing (HPC)	15%
Chiplets	8%

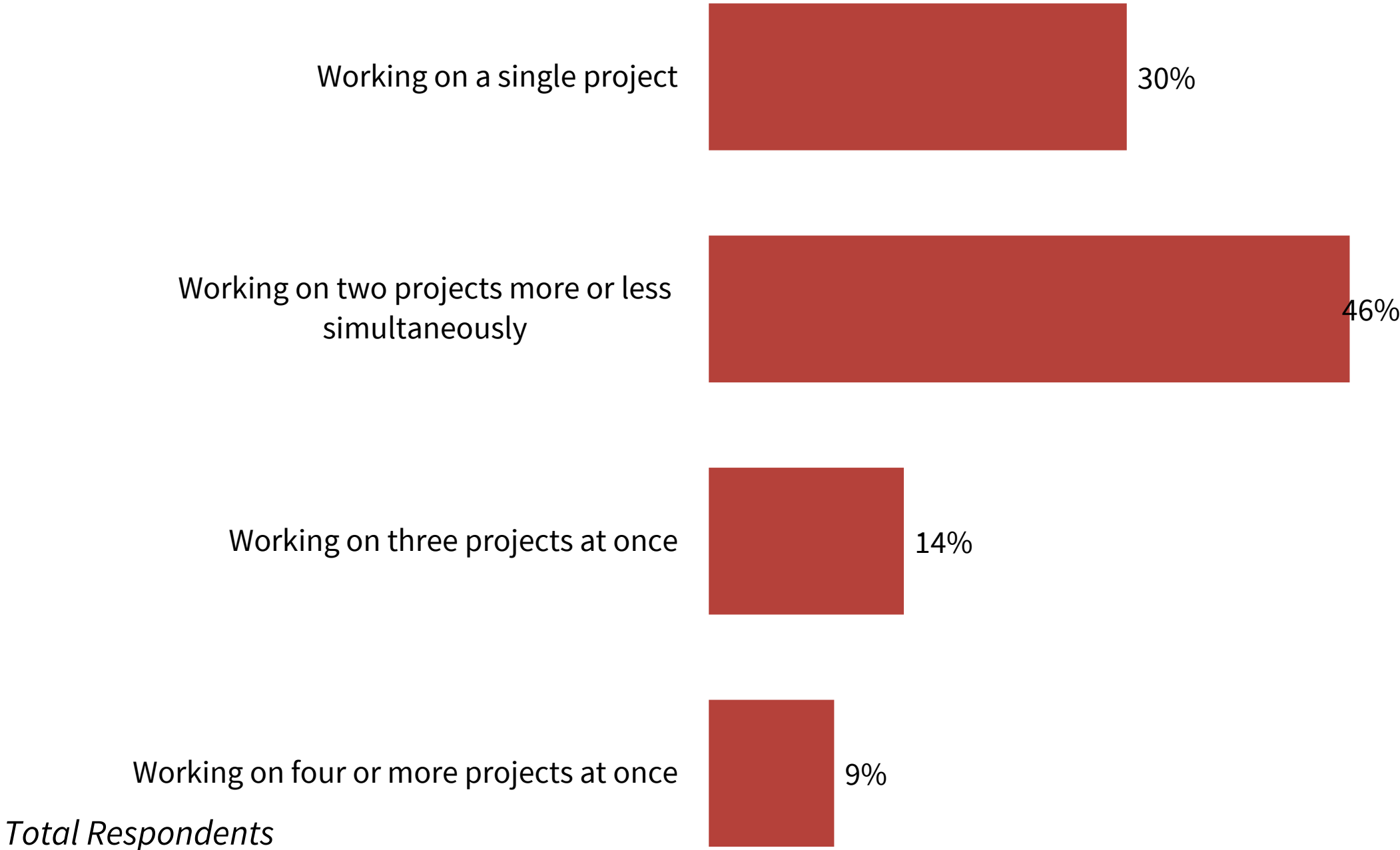*Multiple responses allowed*
Other = 4%

Total Respondents

Embedded development teams have burgeoning workloads

Typical embedded projects take 8 months on average to complete (and longer in larger OEMs and with "new" projects)
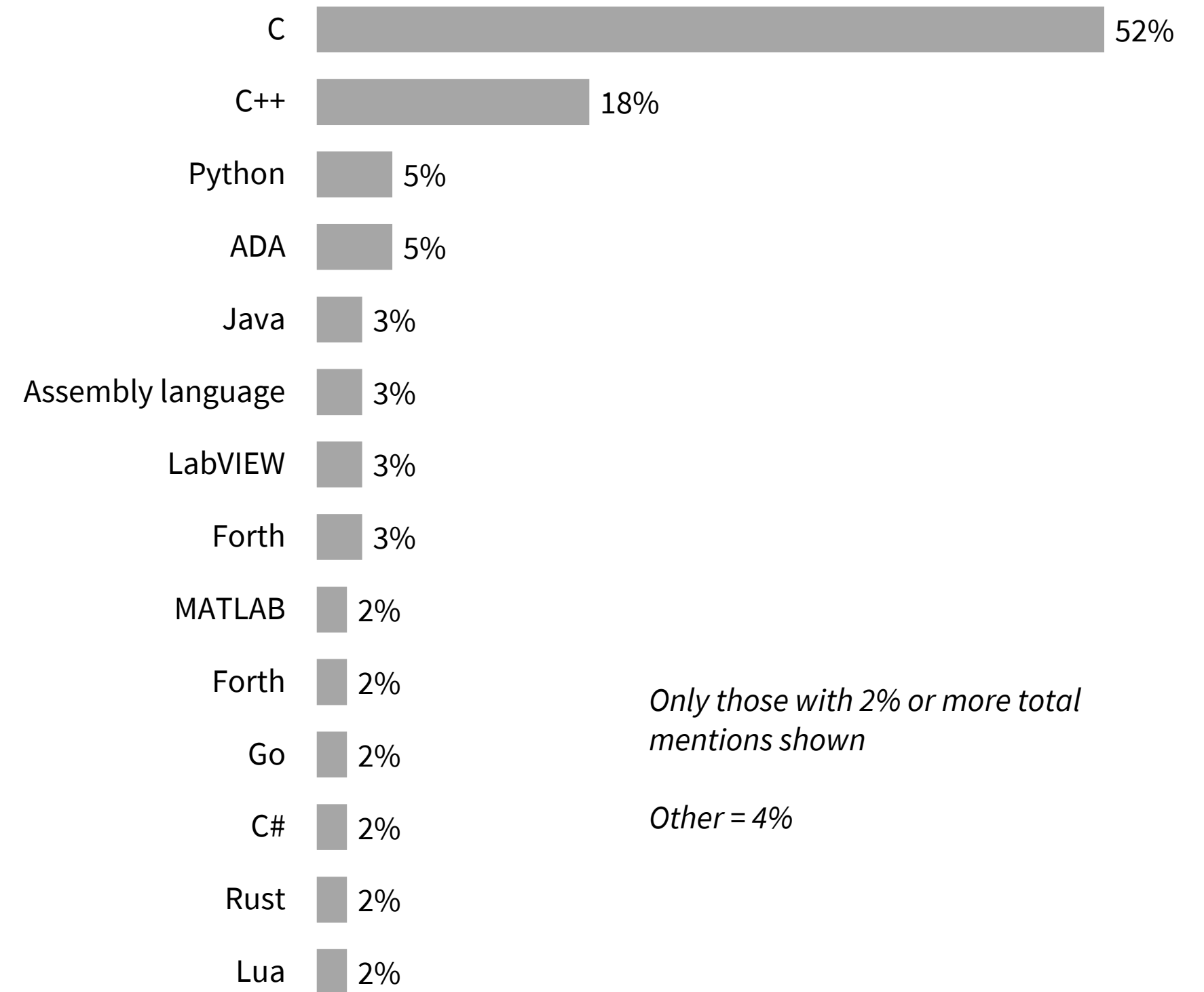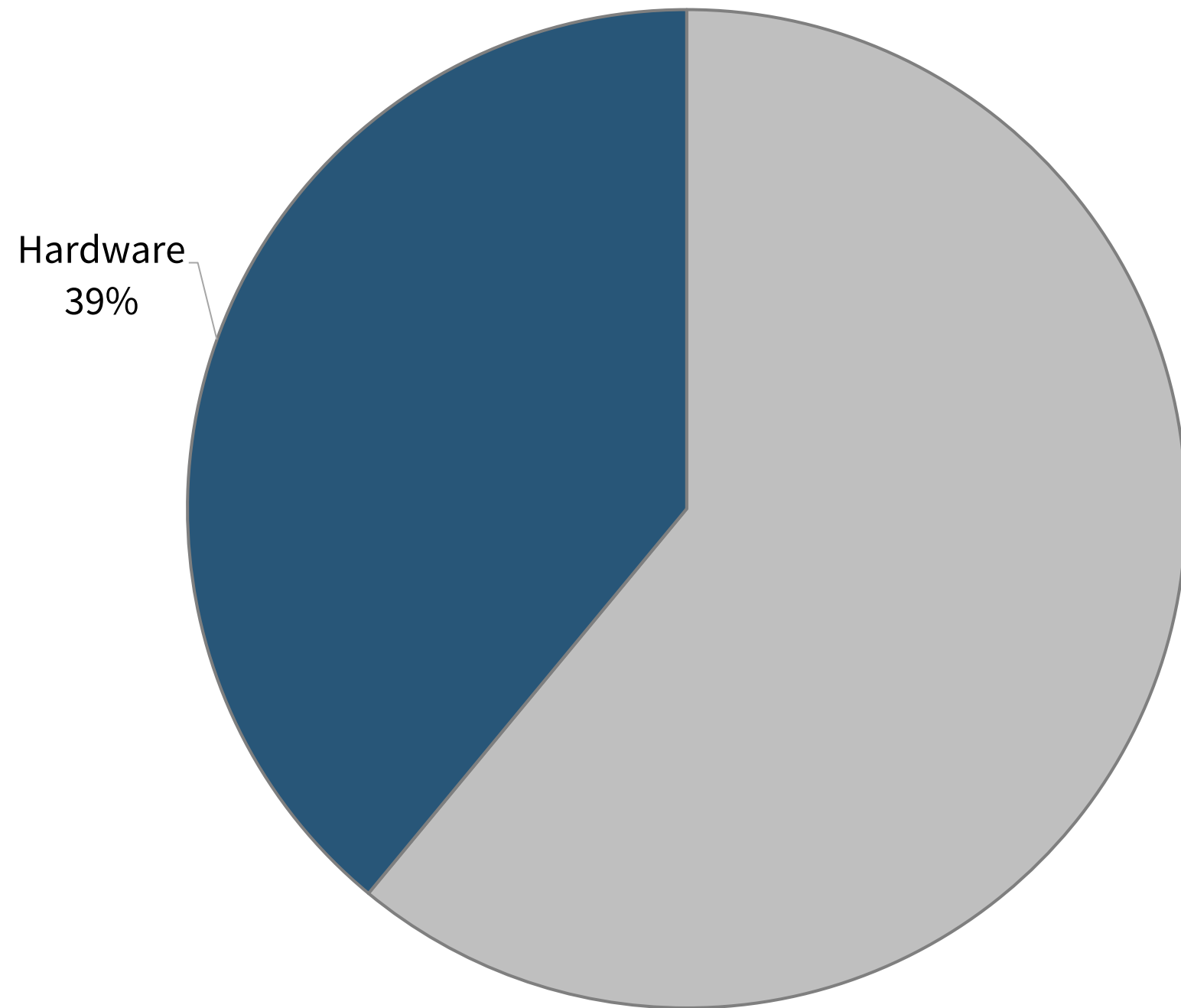
Embedded projects in past year
Average = 4.2 projects

1 project	12%
4 projects	23%
2 projects	23%
3 projects	13%
5 or more projects	27%

Current embedded projects
Average = 2.1 projects

Working on a single project	30%
Working on two projects more or less simultaneously	46%
Working on three projects at once	14%
Working on four or more projects at once	9%

Total Respondents

embedded survey✓

8. **In the past year, how many embedded projects have you worked on?**
9. **How many embedded projects are you working on right now?**

ASPENCORE | 8

Software development requires more cycle time

"C" dominates other languages for embedded software programming

Hardware
39%

Software
61%

C	52%
C++	18%
Python	5%
ADA	5%
Java	3%
Assembly language	3%
LabVIEW	3%
Forth	3%
MATLAB	2%
Forth	2%
Go	2%
C#	2%
Rust	2%
Lua	2%

Only those with 2% or more total mentions shown

Other = 4%

Total Respondents

10. **What is your development team's ratio of total resources (including time/dollars/manpower) spent on software vs. hardware for your embedded projects?**
16. **My current embedded project is programmed mostly in:**

embedded survey✓

ASPENCORE | 9

Embedded development teams are large and cover multiple disciplines

Nearly 20 engineers on the team (more in Americas, fewer in EMEA) – with a plurality focused on software/firmware development
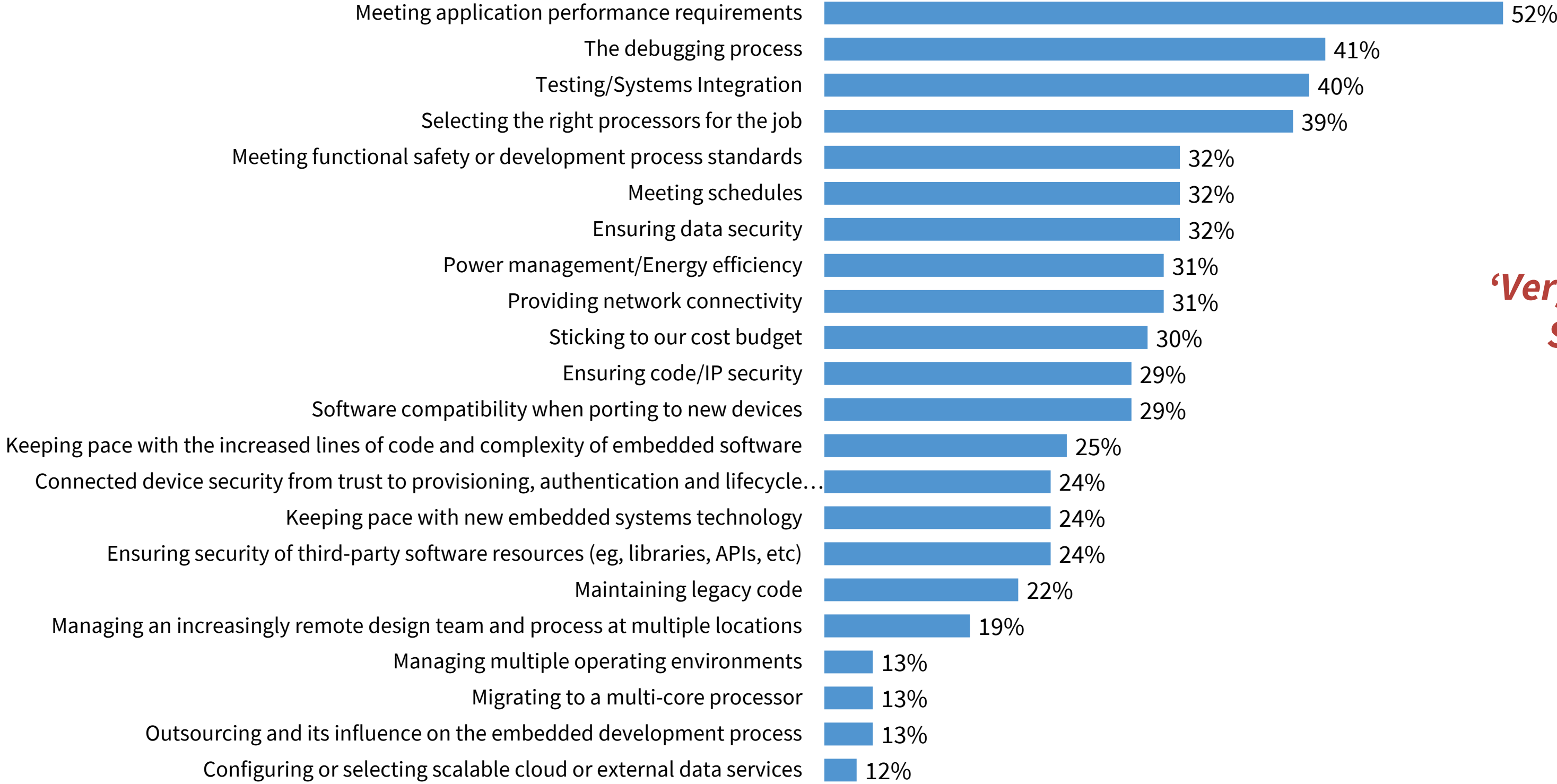
Non-firmware software engineers ████████████████████ 4.3

Firmware engineers ████████████████ 3.6

Hardware engineers ███████████████ 3.4
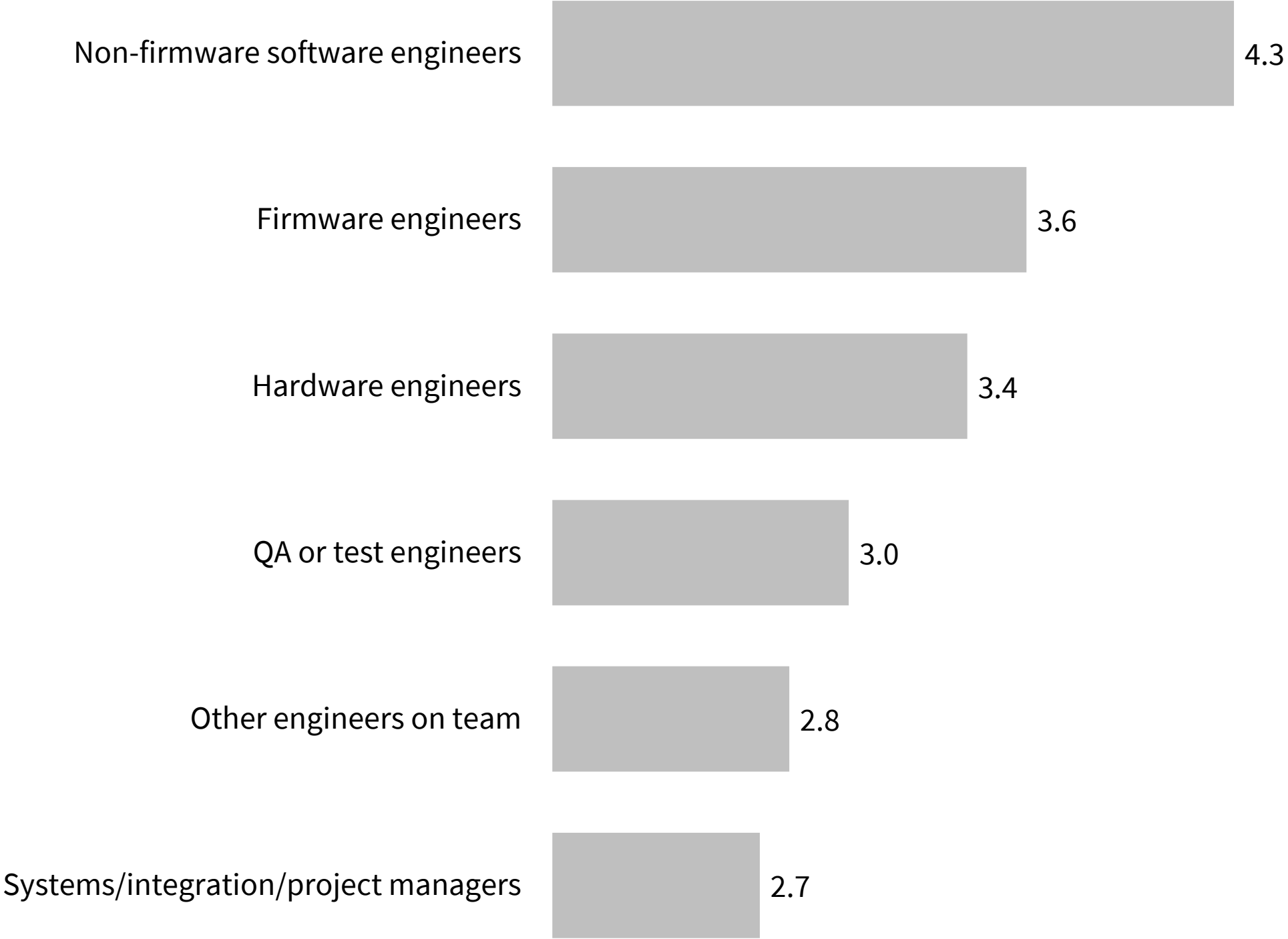
Average engineers per project = 19.8

Mean Scores

QA or test engineers █████████████ 3.0

Other engineers on team ████████ 2.8

Systems/integration/project managers ████████ 2.7

Total Respondents

Meeting performance specs, processor choice and test/debugging are critical issues

Safety, security and power management are also high on the agenda (especially for EMEA and APAC designers)

Meeting application performance requirements	52%
The debugging process	41%
Testing/Systems Integration	40%
Selecting the right processors for the job	39%
Meeting functional safety or development process standards	32%
Meeting schedules	32%
Ensuring data security	32%
Power management/Energy efficiency	31%
Providing network connectivity	31%
Sticking to our cost budget	30%
Ensuring code/IP security	29%
Software compatibility when porting to new devices	29%
Keeping pace with the increased lines of code and complexity of embedded software	25%
Connected device security from trust to provisioning, authentication and lifecycle…	24%
Keeping pace with new embedded systems technology	24%
Ensuring security of third-party software resources (eg, libraries, APIs, etc)	24%
Maintaining legacy code	22%
Managing an increasingly remote design team and process at multiple locations	19%
Managing multiple operating environments	13%
Migrating to a multi-core processor	13%
Outsourcing and its influence on the embedded development process	13%
Configuring or selecting scalable cloud or external data services	12%

'Very Important' Summary

Total Respondents

Better debugging and SW tools, improved safety and security and power join processor selection as most critical design challenges

Challenge	Percentage
Improving the debugging process	32%
Functional safety	31%
Security concerns	31%
Processors	30%
Software tools	29%
Dealing with low power	29%
Managing increases in code size and complexity	26%
Dealing with wireless	25%
OS/RTOS	24%
IDE	23%
Integrating new technology or tools	22%
Hardware tools	21%
SoCs/ASICs/ASSPs	20%
Programmable logic	20%
Connecting to the cloud	19%
Memories (random and non-volatile)	18%
Buses/interconnects	16%
Integrating external IP into your designs	12%

APAC design teams are especially concerned about nearly all these issues

'Very Important' Summary

Total Respondents

embedded survey✓

20. **Thinking about next year, how important will the following technology challenges be regarding future embedded designs?**

ΛSPENCORE | 12

Reuse of software code, hardware and HW IP is the norm

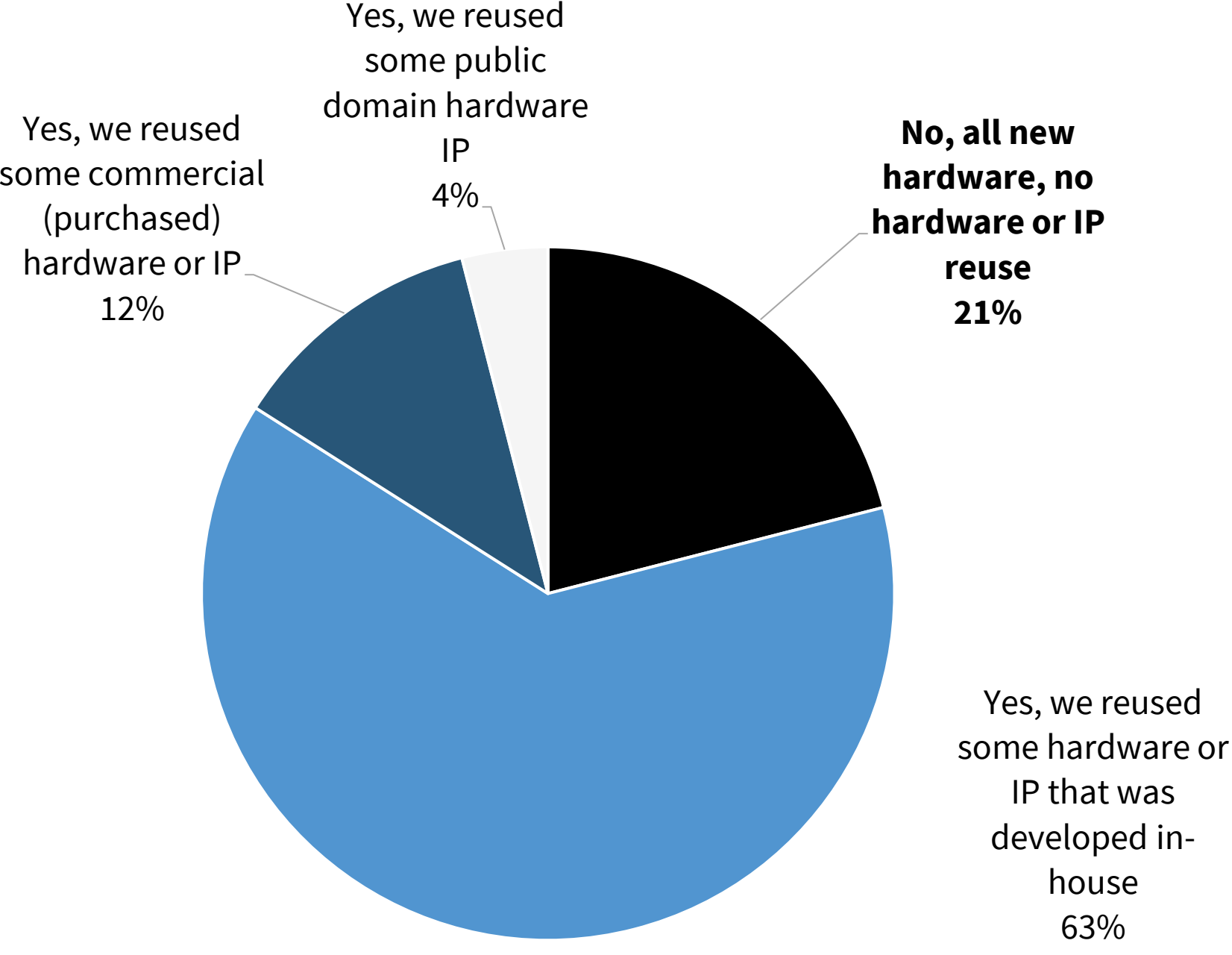Proprietary software code reuse is somewhat more common than hardware or IP reuse

Software Reuse = 89%
2019 = 88%

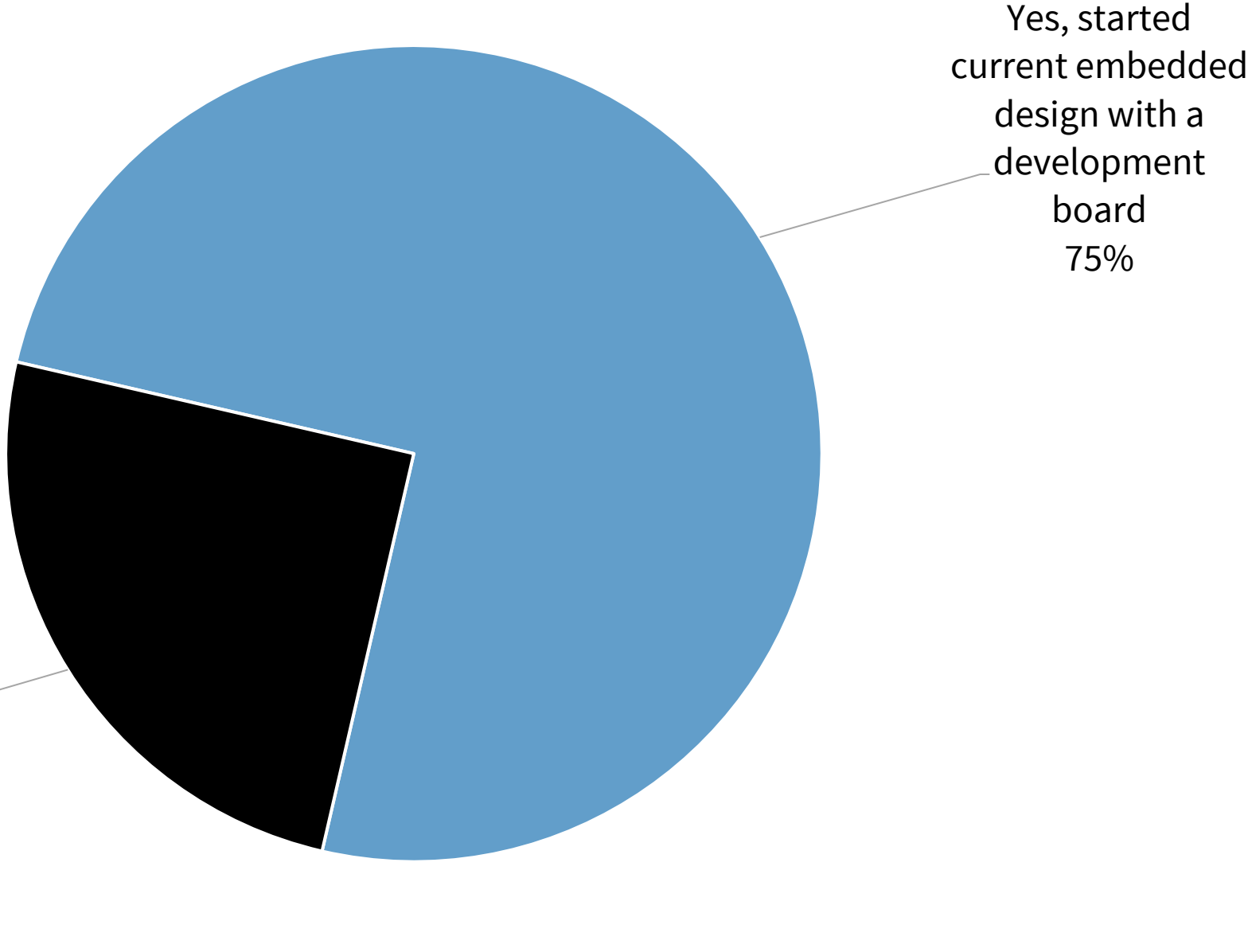Yes, we reused some open-source, public domain, or shareware code
10%

No, all new software, no code reuse
11%

Yes, we reused some commercial (purchased) code
5%

Yes, we reused some code developed in-house
74%
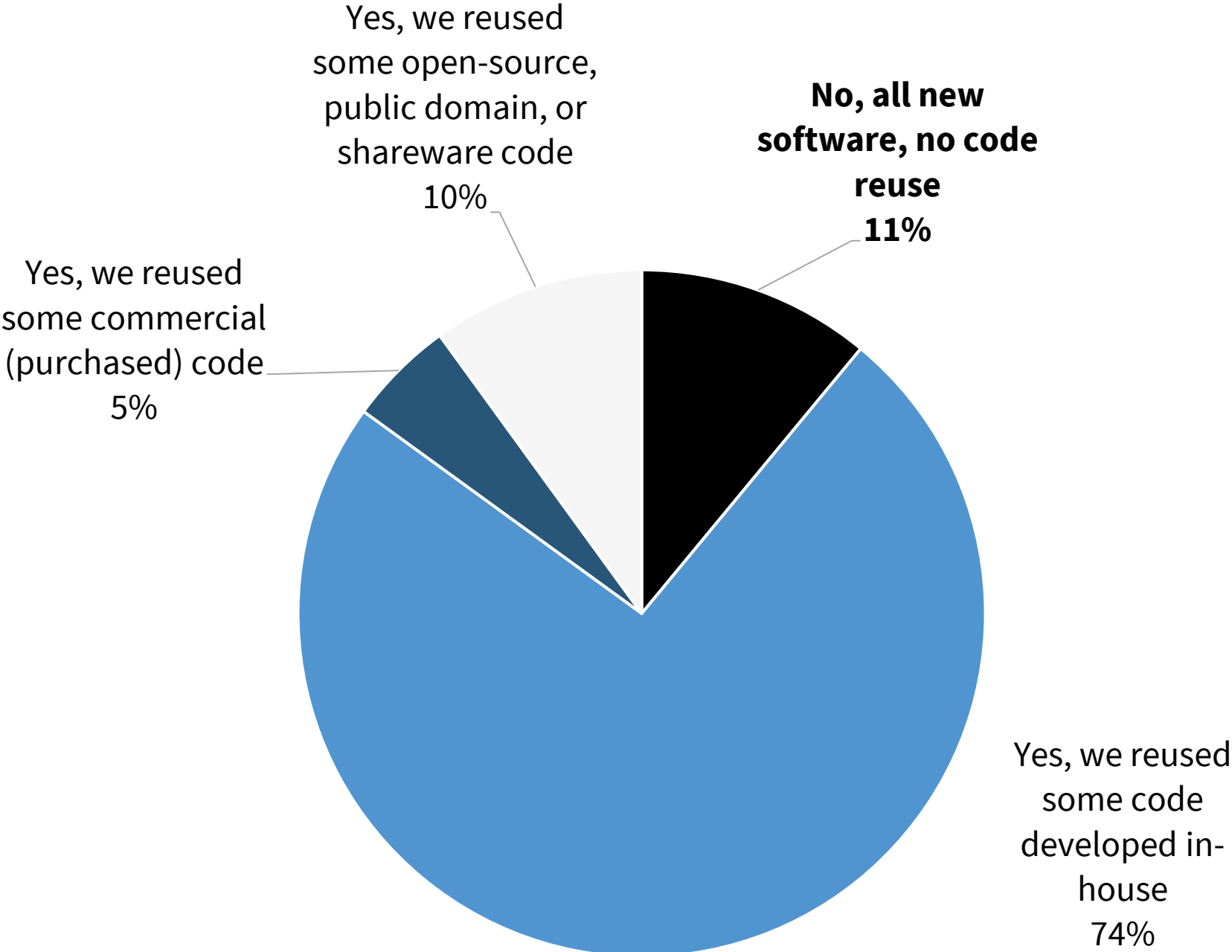
Hardware Reuse = 79%
2019 = 77%

Yes, we reused some public domain hardware IP
4%

Yes, we reused some commercial (purchased) hardware or IP
12%

No, all new hardware, no hardware or IP reuse
21%

Yes, we reused some hardware or IP that was developed in-house
63%

Total Respondents

embedded survey✔

17. Does your current project reuse code from a previous embedded project?
18. Does your current embedded project reuse hardware or hardware IP from a previous project?

ASPENCORE | 13

Use of development boards in embedded design is widespread

Among those using development boards, more than half use *Raspberry Pi* and *Arduino*

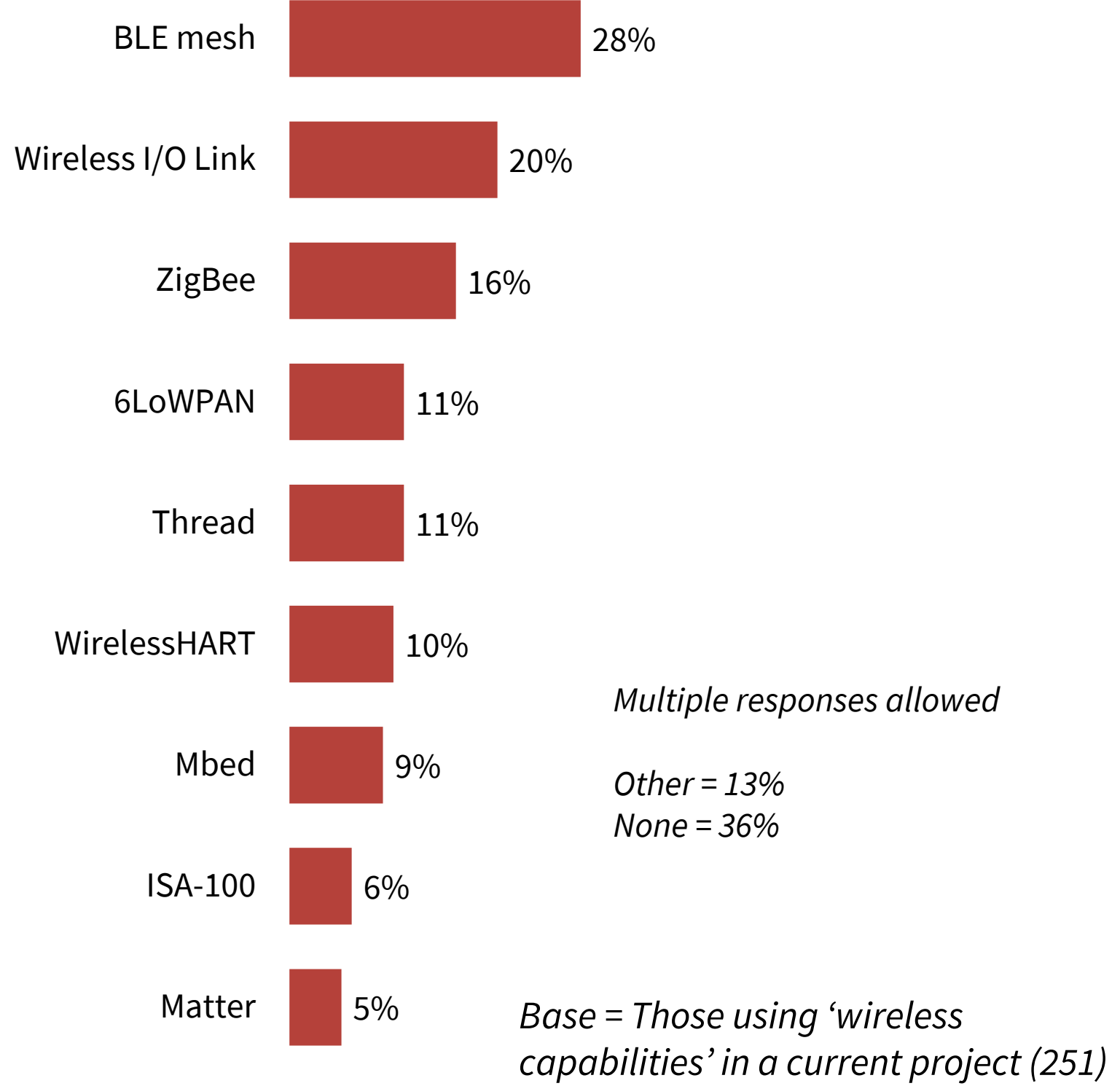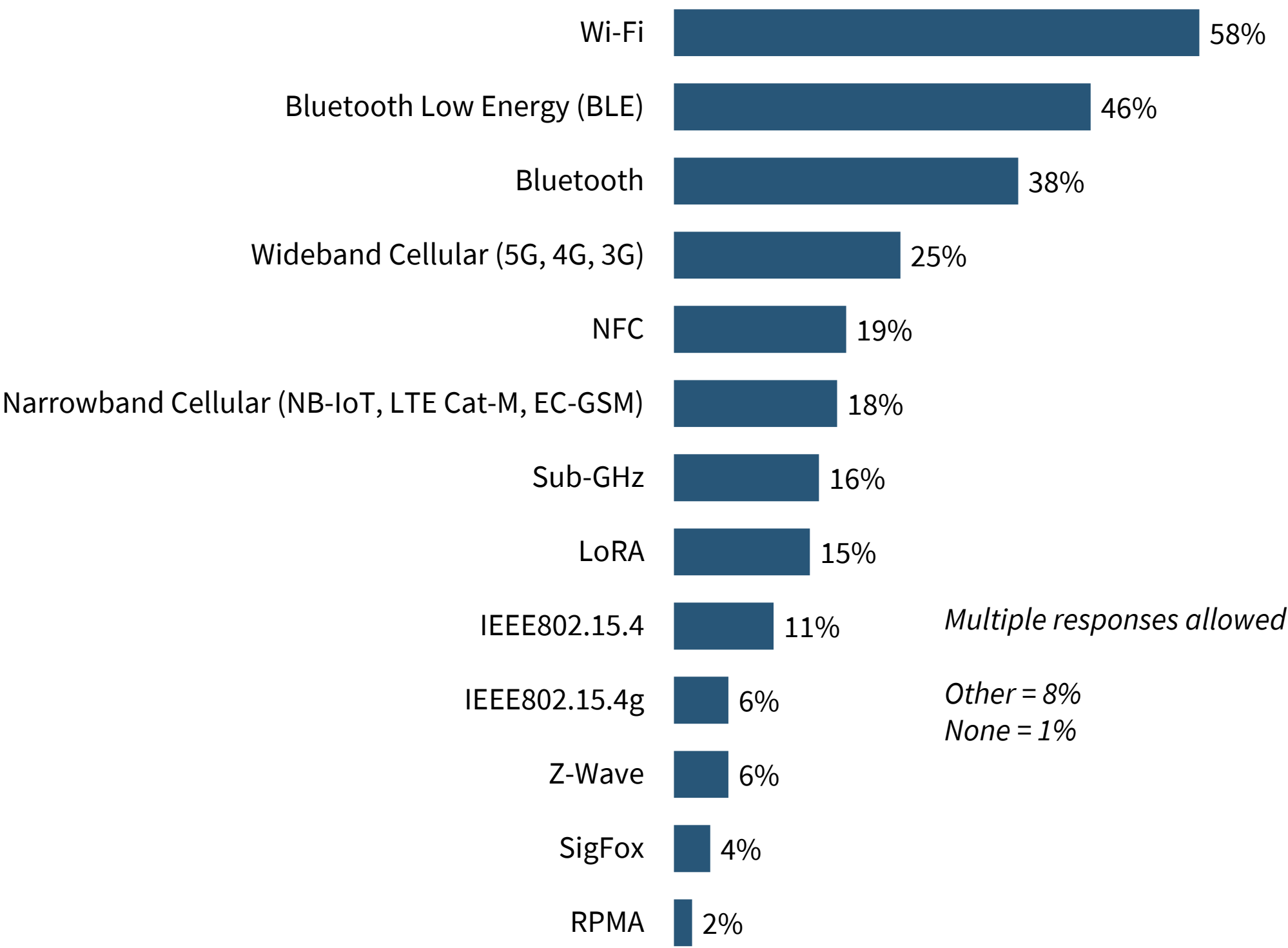## Development Board Use when Starting Embedded Projects

Yes, started current embedded design with a development board
75%

No, did not start current embedded design with a development board
25%

Total Respondents

Board Used in Current Design(s)

Board	%
Raspberry Pi	26%
Arduino	26%
Custom design	21%
Proprietary	12%
Beagle Board/Bone	10%
LaunchPad	7%
COM Express	7%
Mbed	5%
PCI Express (PCIe)	5%
ATX	4%
CompactPCI	4%
COM	4%
COM-HPC	4%
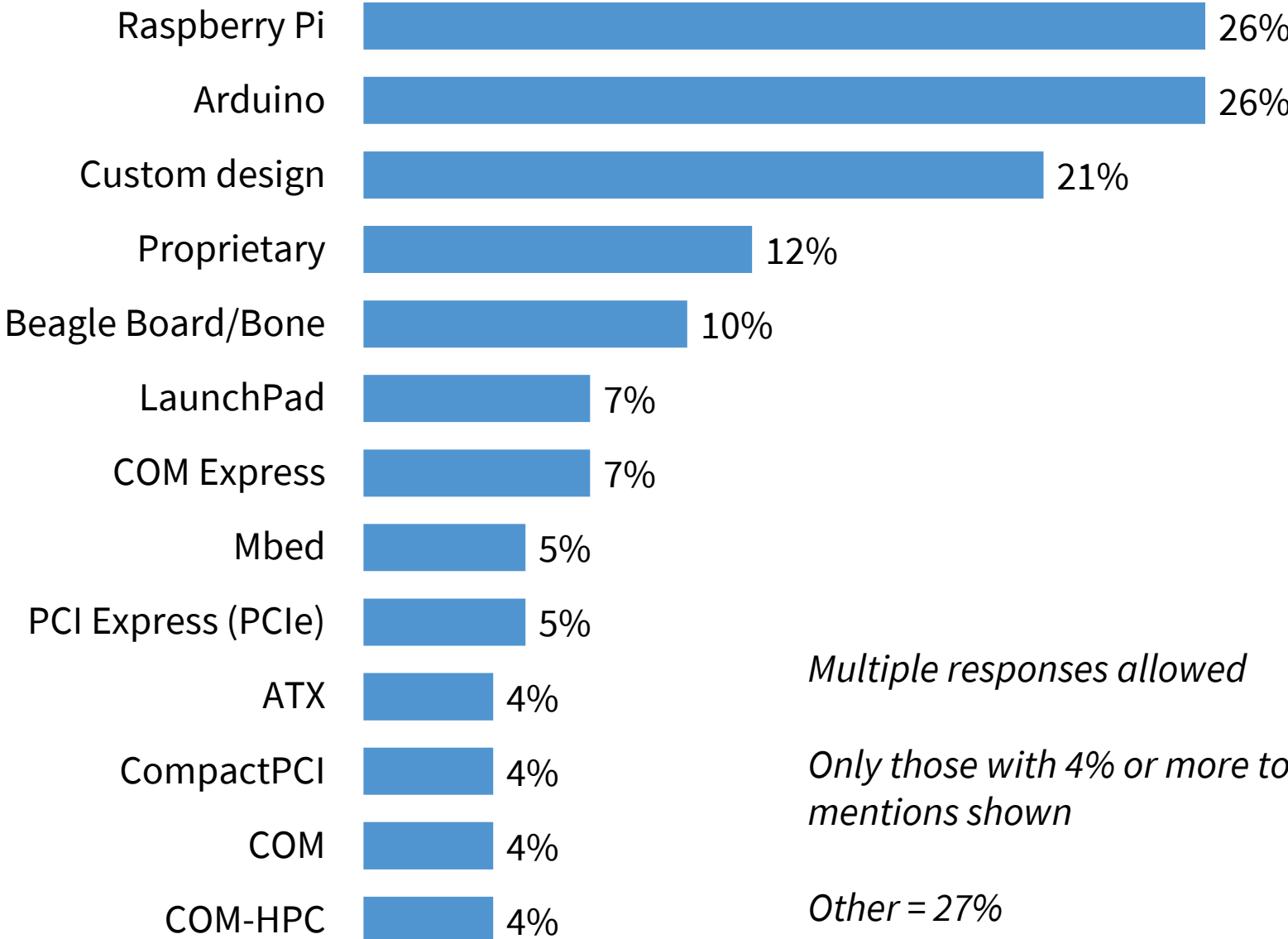
Multiple responses allowed

Only those with 4% or more total mentions shown

Other = 27%
None = 7%

Base = Those using development board 488()

embedded survey
12. Did you start your current embedded design with a development board?
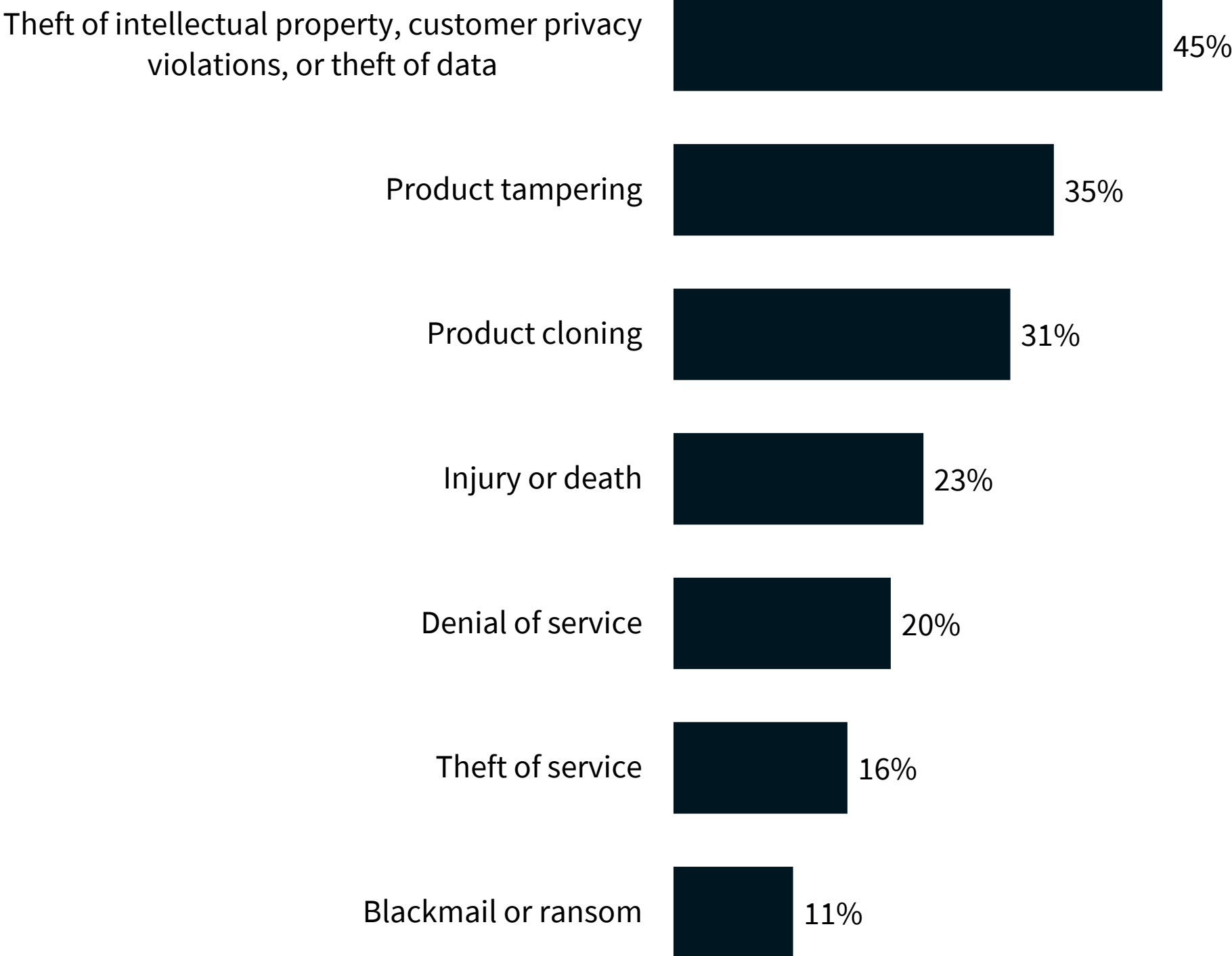13. Which development boards are you currently using?
ASPENCORE | 14

Over one-third of embedded designs incorporate wireless capabilities

Wi-Fi and Bluetooth are the most used interfaces and Bluetooth Low Energy mesh the most popular protocol

Interface	%
Wi-Fi	58%
Bluetooth Low Energy (BLE)	46%
Bluetooth	38%
Wideband Cellular (5G, 4G, 3G)	25%
NFC	19%
Narrowband Cellular (NB-IoT, LTE Cat-M, EC-GSM)	18%
Sub-GHz	16%
LoRA	15%
IEEE802.15.4	11%
IEEE802.15.4g	6%
Z-Wave	6%
SigFox	4%
RPMA	2%

Multiple responses allowed

Other = 8%
None = 1%

Protocol	%
BLE mesh	28%
Wireless I/O Link	20%
ZigBee	16%
6LoWPAN	11%
Thread	11%
WirelessHART	10%
Mbed	9%
ISA-100	6%
Matter	5%

Multiple responses allowed

Other = 13%
None = 36%

Base = Those using 'wireless capabilities' in a current project (251)

embedded survey ✓

4. What wireless interfaces does your current project include?
5. What wireless protocols/stacks does your current project include?

ASPENCORE | 15

Embedded design requires attention to security

IP theft, product tampering, and cloning are primary issues, especially for larger OEMs

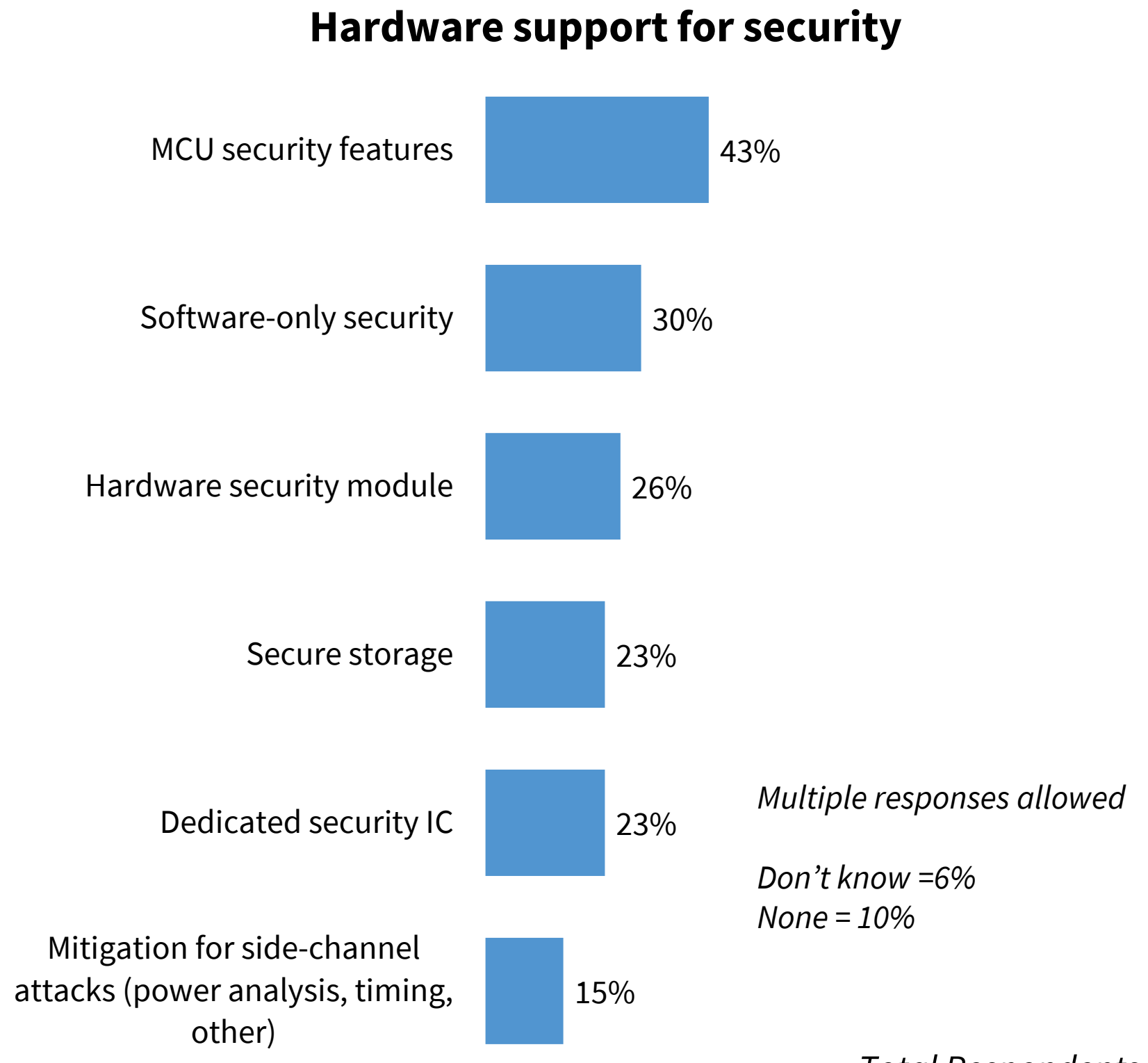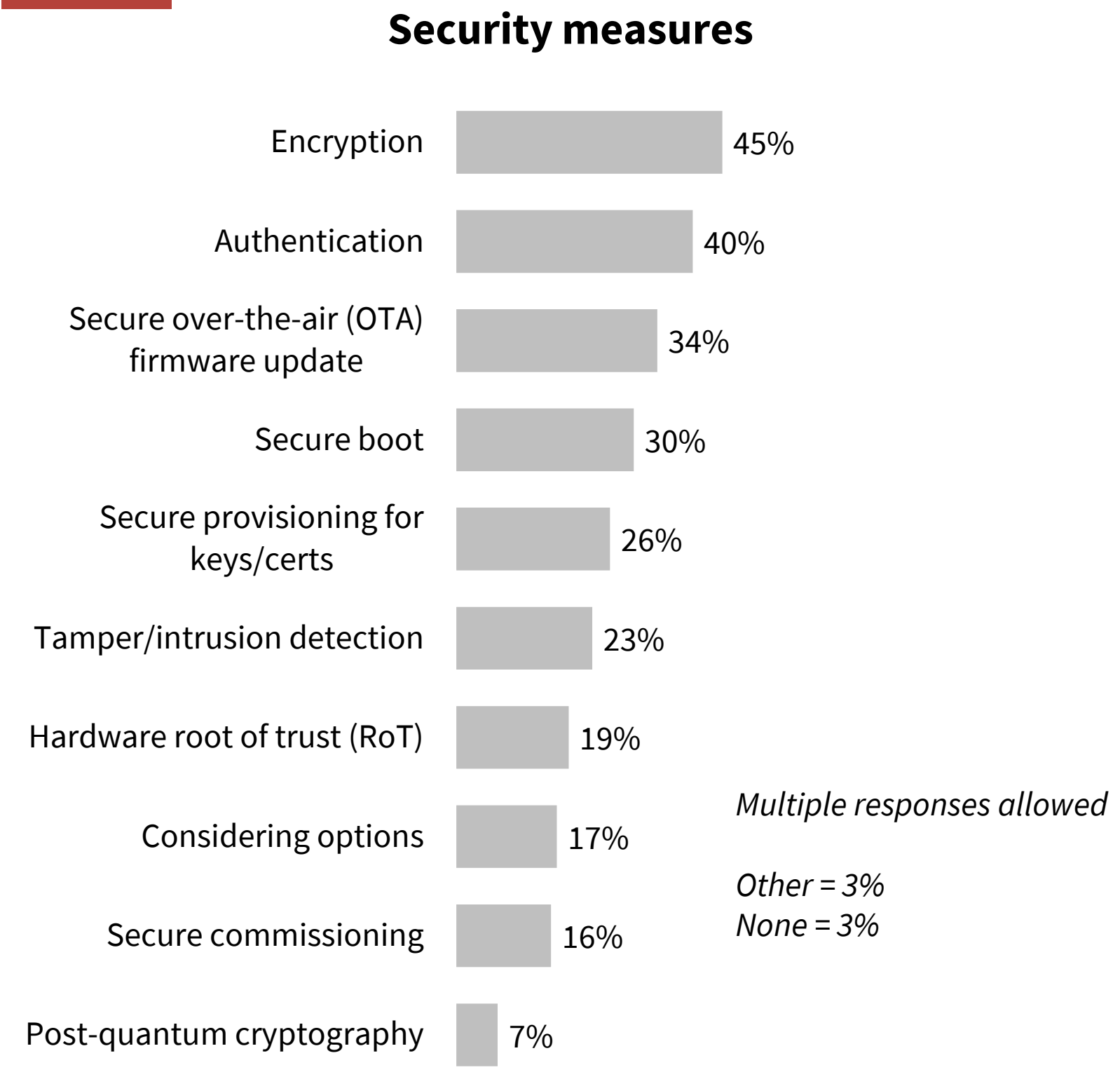| Category	Percentage
Theft of intellectual property, customer privacy violations, or theft of data	45%
Product tampering	35%
Product cloning	31%
Injury or death	23%
Denial of service	20%
Theft of service	16%
Blackmail or ransom	11%

Multiple responses allowed

Other = 2%
None = 9%

Total Respondents

Embedded design teams utilize multiple security solutions

Encryption, authentication and secure "over the air" firmware updates, along with MCU security features are most relied on
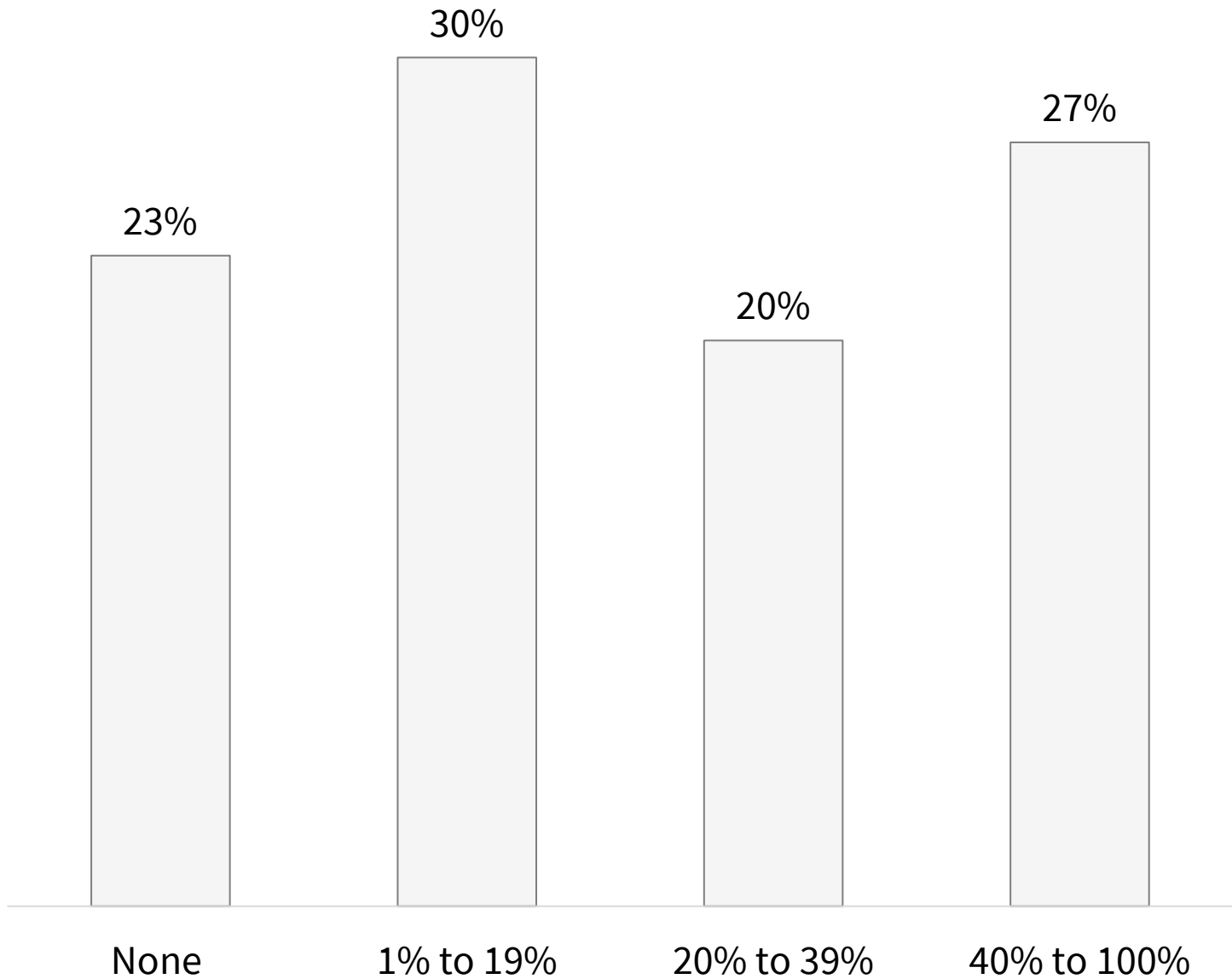
Security measures
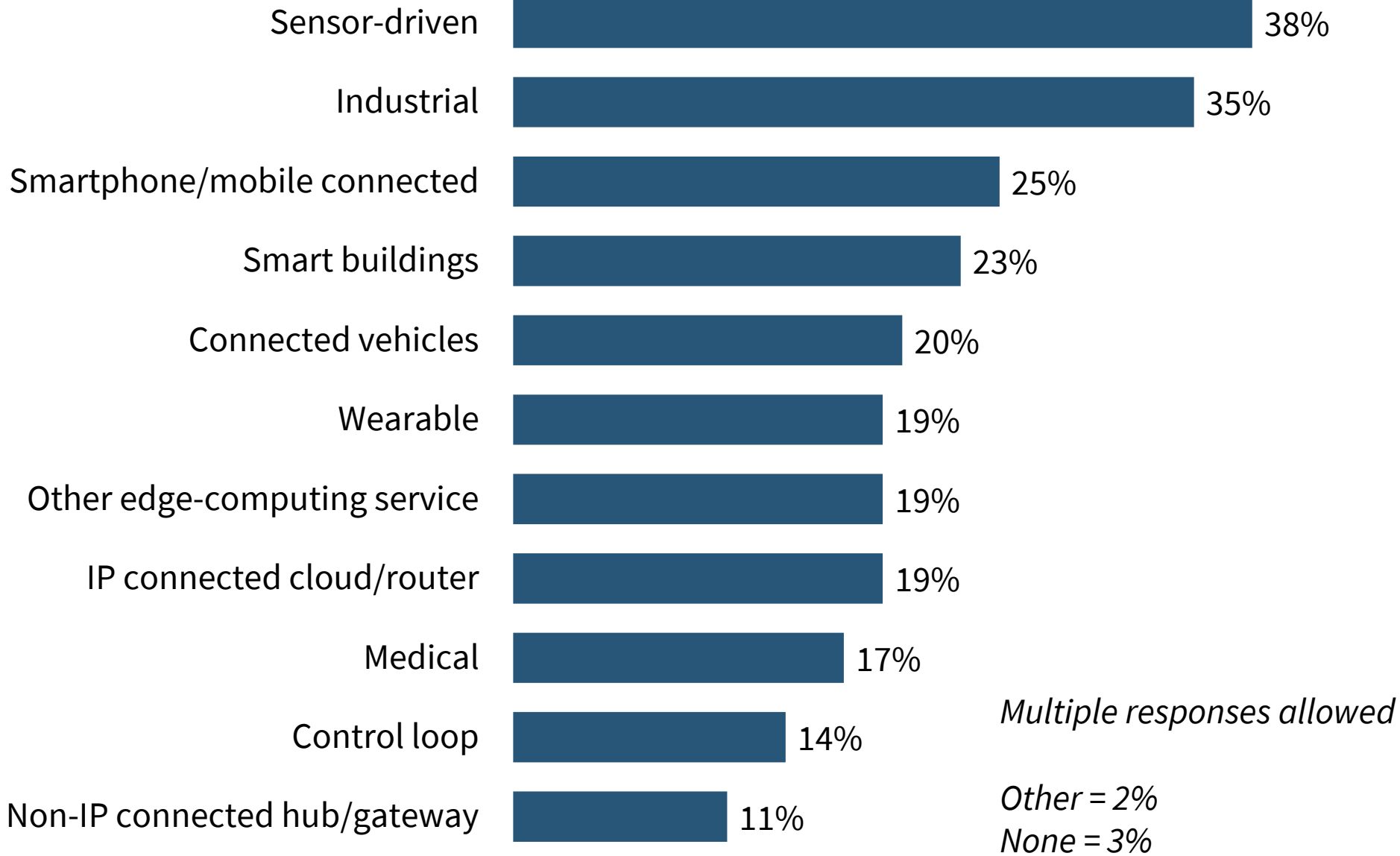
Encryption	45%
Authentication	40%
Secure over-the-air (OTA) firmware update	34%
Secure boot	30%
Secure provisioning for keys/certs	26%
Tamper/intrusion detection	23%
Hardware root of trust (RoT)	19%
Considering options	17%
Secure commissioning	16%
Post-quantum cryptography	7%

Multiple responses allowed

Other = 3%
None = 3%

Hardware support for security

MCU security features	43%
Software-only security	30%
Hardware security module	26%
Secure storage	23%
Dedicated security IC	23%
Mitigation for side-channel attacks (power analysis, timing, other)	15%

Multiple responses allowed

Don't know =6%
None = 10%

Total Respondents

22. What security measures are you incorporating into your current design?
23. What hardware support for security are you incorporating?

ASPENCORE | 17

Internet of Things (IoT) continues to attract attention

Nearly one-third of embedded design is devoted wholly or partially to IoT, most for sensor-driven, industrial or mobile communications applications

Projects devoted to IoT
Mean = 29% of projects
2019 = 21% of projects

None: 23%
1% to 19%: 30%
20% to 39%: 20%
40% to 100%: 27%

Total Respondents

Types of Applications for IoT

Sensor-driven: 38%
Industrial: 35%
Smartphone/mobile connected: 25%
Smart buildings: 23%
Connected vehicles: 20%
Wearable: 19%
Other edge-computing service: 19%
IP connected cloud/router: 19%
Medical: 17%
Control loop: 14%
Non-IP connected hub/gateway: 11%

Multiple responses allowed

Other = 2%
None = 3%

Base = Those primary devoted to IoT applications (870)

24. **In the coming year, approximately what percentage of your projects will be primarily devoted to Internet of Things (IoT) applications or devices?**
25. **If you are developing Internet of Things (IoT) applications, please indicate the types of applications.**

ΛSPENCORE | 18

Embedded development makes use of advanced technology capabilities

Embedded AI and machine learning attract the most attention, followed by embedded vision and speech capabilities
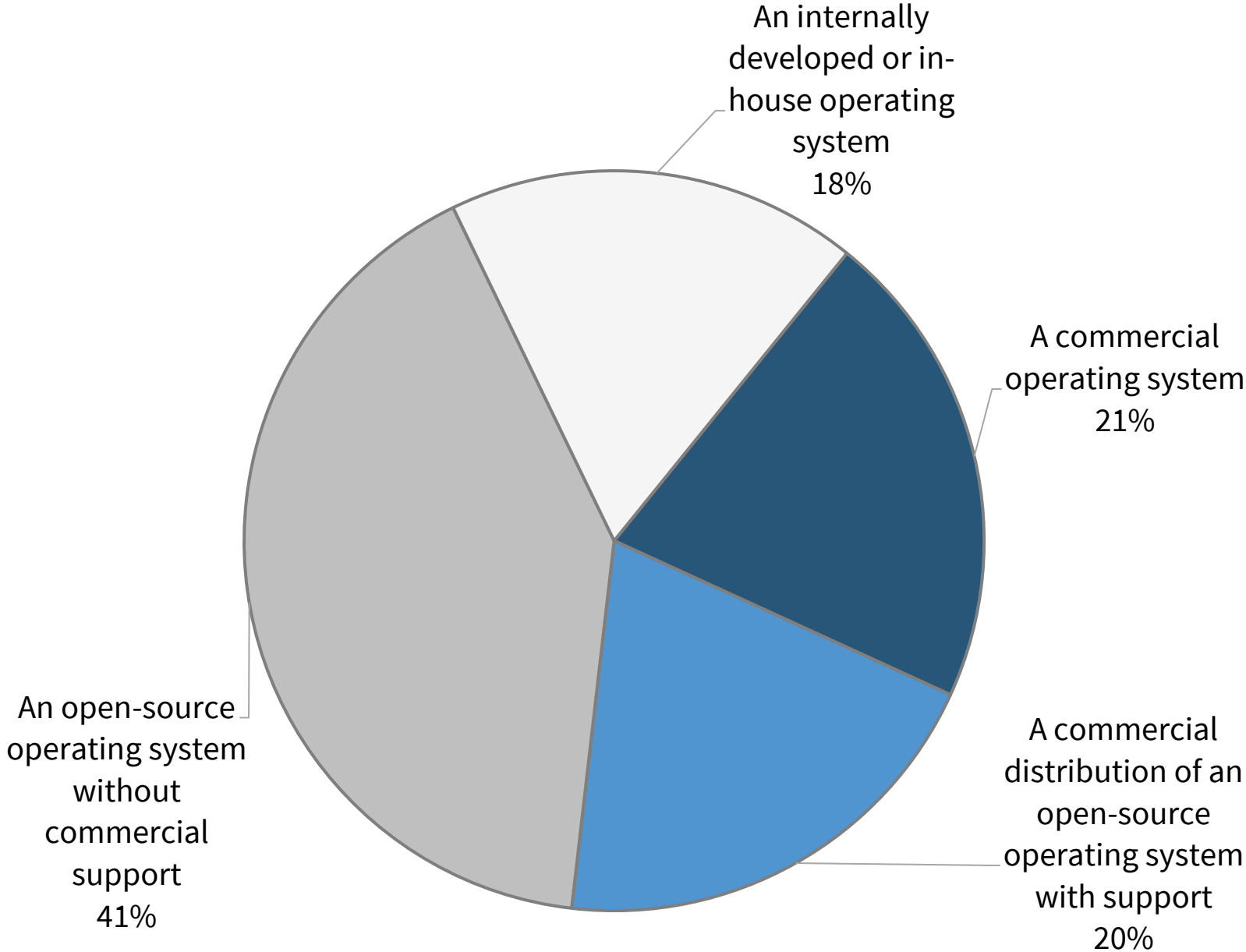
Total Interest 50%   47%   36%   29%   26%   21%   18%

Considering

| | | | | | | |
| 24% | 24% | 19% | 15% | 6% | 11% | 8% |

Currently Using

| | | | | | | |
| 26% | 23% | 17% | 14% | 20% | 10% | 10% |

| Embedded AI | Machine learning model-based capabilities | Embedded vision | Embedded speech | Other AI/cognitive capabilities | Augmented Reality (AR) capabilities | Virtual Reality (VR) capabilities |

Total Respondents

27. Which of the following advanced technologies are you <u>currently using</u> in your embedded systems?
28. Which of the following advanced technologies are you <u>considering using</u> in your future embedded systems?

ASPENCORE | 19

embedded survey✓

Operating Systems

Most embedded projects utilize an operating system

Although open source is popular, four in ten use either commercial OS or open-source OS distributed commercially

74% use an OS in current embedded project

OS Used in Current Embedded Project

An internally developed or in-house operating system
18%

A commercial operating system
21%

A commercial distribution of an open-source operating system with support
20%

An open-source operating system without commercial support
41%

Total Respondents

Reasons for not using commercial OS

Current solution works fine	58%
Want to avoid reliance on commercial supplier	42%
Commercial alternatives are too expensive	38%
Security concerns with commercial alternatives	18%
Too much trouble to learn to use commercial alternative	16%
Safety concerns with commercial alternatives	14%
Commercial alternatives use too much memory	13%
Incompatible with existing software, applications, or drivers	12%
Commercial alternatives lack an important feature I need	11%
No need for real-time	10%
No need for multithreading or multitasking	9%

Multiple responses allowed

Other = 6%

Base = Those not using commercial OS (284)

29. Does your current embedded project use an operating system, RTOS, kernel, software executive, or scheduler of any kind?
30. My current embedded project uses:
31. 31. What are your reasons for NOT currently using a commercial operating system?

ASPENCORE | 21

embedded survey

OS use will increase, but open-source share will grow

Nearly 30% of those now using commercial OS are considering open-source alternatives

86% will use an OS for next embedded project

An internally developed or in-house operating system
12%

A commercial distribution of an open-source operating system with support
21%

A commercial operating system
16%

An open-source operating system without commercial support
37%

Will not use an operating system
14%
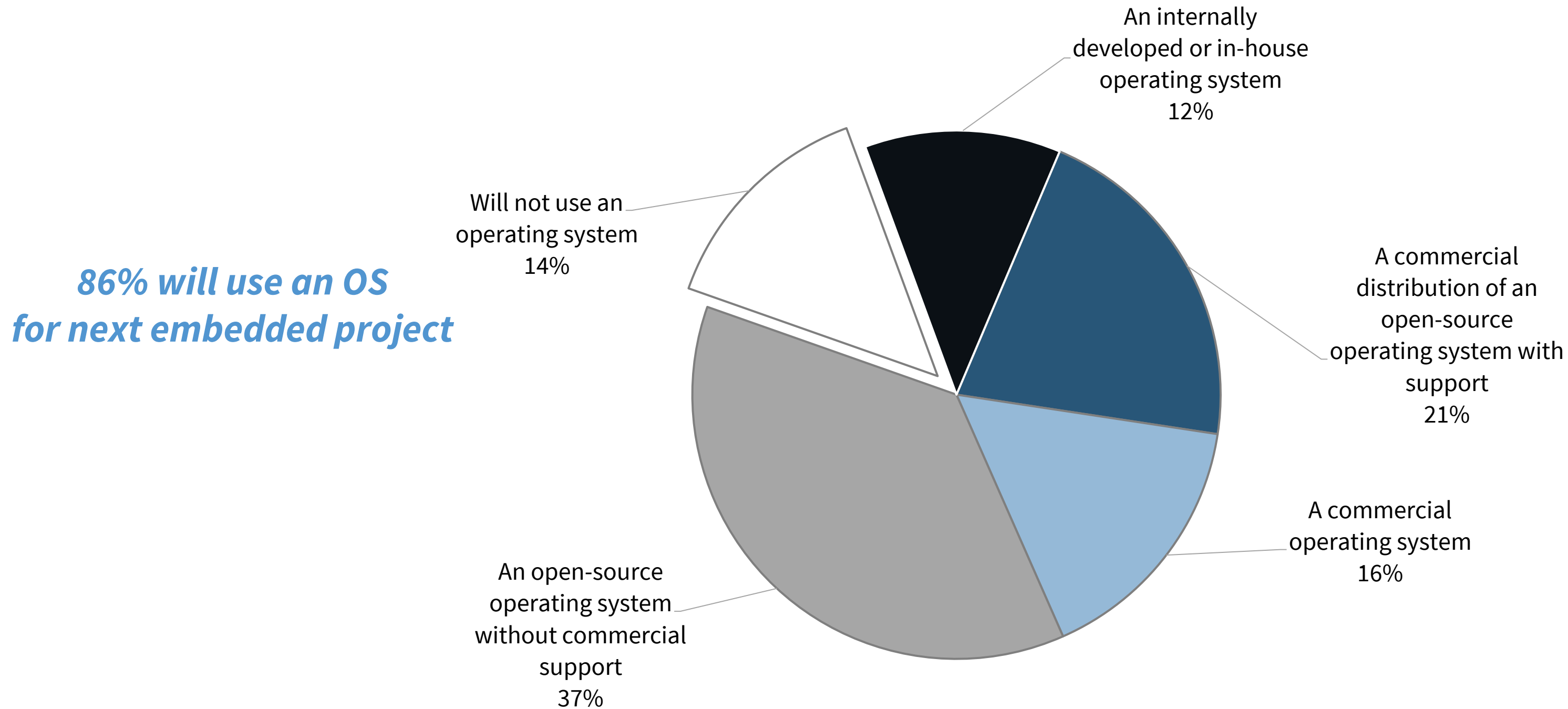
Total Respondents

Most popular embedded OSs – Embedded Linux, FreeRTOS and Ubuntu

Top 3 OSs are especially popular in APAC, while Embedded Linux is used more in the Americas

OS	Percentage
Embedded Linux	44%
FreeRTOS	44%
Ubuntu	18%
Debian (Linux)	17%
Android	16%
In-house/custom	11%
Keil (RTX)	9%
Express Logic (ThreadX)	8%
Texas Instruments RTOS	7%
Red Hat (IX Lunix)	6%
QNX (QNX)	6%
AnalogDevices (VDK)	5%
ECos	5%
Microsoft (Windows 7 Compact or earlier)	4%
Freescale MQX	4%
Micrium (uC/OS-II)	4%
Texas Instruments (DSP/BIOS)	4%
Wind River (VxWorks)	4%
Micrium (uC/OS-III)	4%
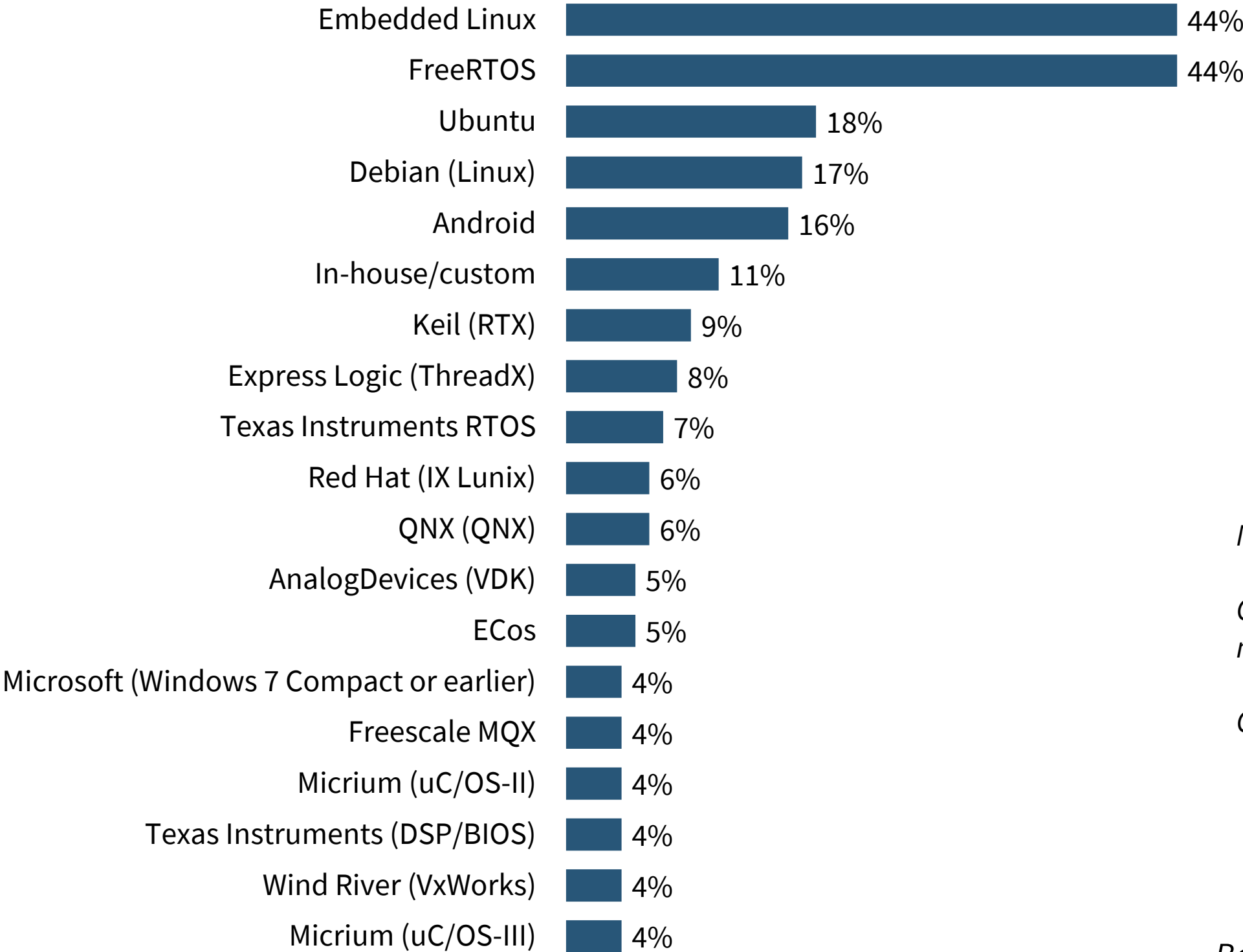
Multiple responses allowed

Only those with 4% or more total mentions shown

Other = 7%

Base = Those who will use an OS (566)

Those using a commercial OS look for documentation, hardware compatibility and support to complement real-time capabilities

Large OEMs and APAC developers put particular emphasis on most commercial OS capabilities

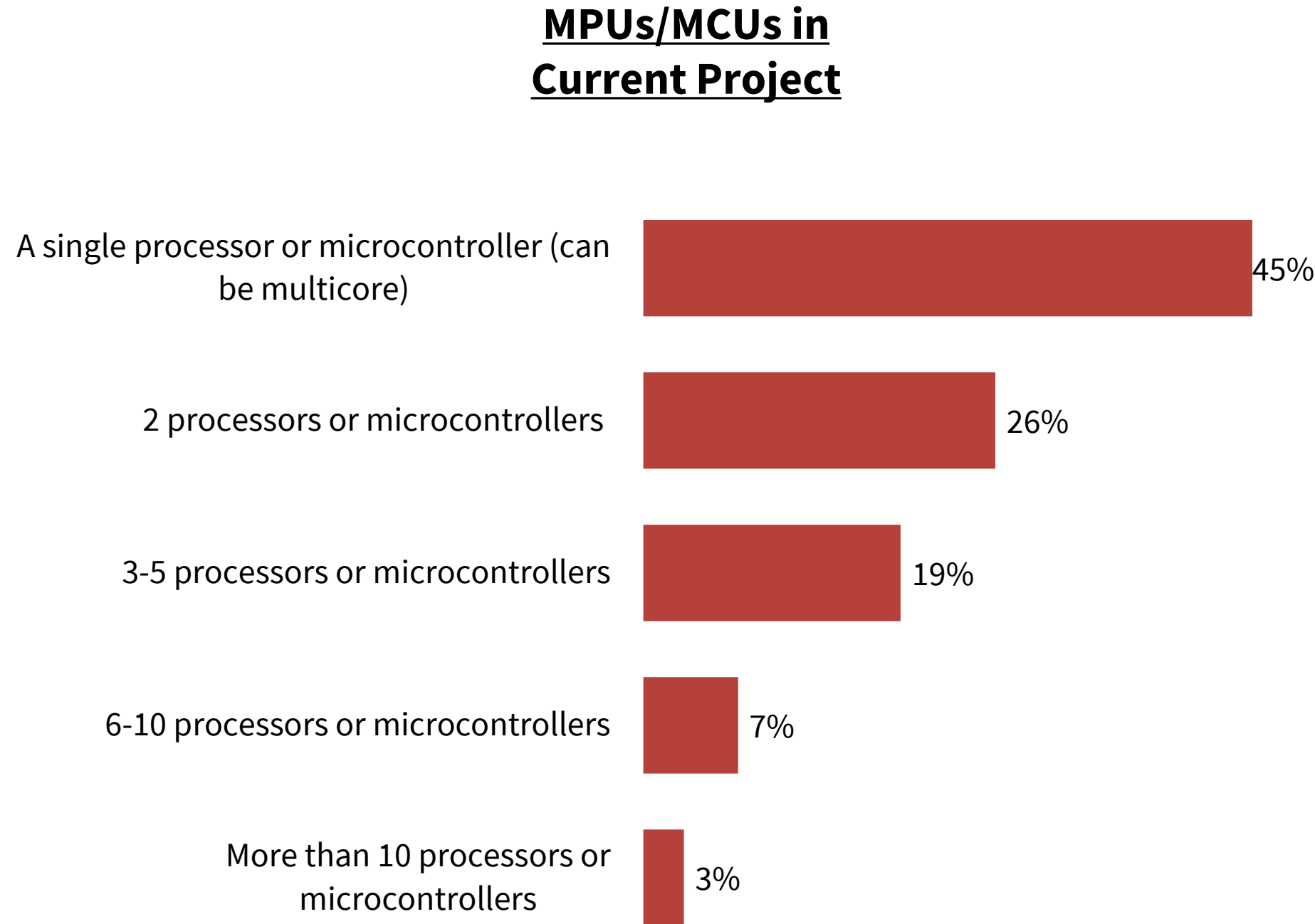| Factor	%
Real-time capability	49%
Documentation	43%
Processor or hardware compatibility	43%
Support for my processor including drivers	43%
Good software tools	40%
Security	37%
Technical support	36%
Ease of future maintenance	35%
Networking capability	34%
Overall cost	34%
Safety Certification	34%
Code size / memory usage	31%
Scheduling efficiency	30%
Modularity	29%
Royalty-free	28%
Context switch time	27%
Power management	26%
Supplier's reputation	26%
Multicore support	25%
Customer's desire	21%
Hypervisor functionality	18%

'Very Important'
Summary

Multiple responses allowed

Base = Those <u>using</u> commercial OS (200)

Microprocessors / Microcontrollers / FPGAs

Most embedded projects use multiple processors

Multiprocessor designs are more common in the Americas.   32-bit processors continue to be the most prevalent.

MPUs/MCUs in Current Project

Category	Percentage
A single processor or microcontroller (can be multicore)	45%
2 processors or microcontrollers	26%
3-5 processors or microcontrollers	19%
6-10 processors or microcontrollers	7%
More than 10 processors or microcontrollers	3%

Main Processor Clock Rate

Clock Rate	Percentage
Under 10 MHz	2%
10 - 24 MHz	8%
25 - 49 MHz	8%
50 - 99 MHz	14%
100 - 249 MHz	27%
250 - 499 MHz	15%
500 - 749 MHz	5%
750 MHz to 999 MHz	5%
1 GHz to 2GHz	12%
2GHz+	4%

Total Respondents

Embedded designers prefer to MPUs/MCUs with which they have experience

Half choose same processor to ensure continuity, but others want access to more features and clearer roadmap

Use of Same Processor from Prior Project

Used same processor
53%

Used different processor
47%

Total Respondents

Reason for Switching Processors

New processor has better features	40%
New processor has better future growth path / roadmap	27%
Previous processor no longer available	23%
Previous processor too slow (needed increased performance/bit width)	18%
New processor has better software / development tools	18%
Not my choice / processor chosen for me	17%
Needed a lower power processor	16%
Previous processor too slow (needed higher clock speed)	13%
Previous processor too expensive	10%
To change operating system	6%
Unhappy with previous processor's supplier	6%

Multiple responses allowed

Other = 2%

54% chose a processor from a different family, architecture or instruction set

Base = Those who did not use the same processor (305)

Processor selection involves weighing many interrelated factors

Performance, available peripherals, HW/SW tool environment, support ecosystem, and other issues are high on developer agendas
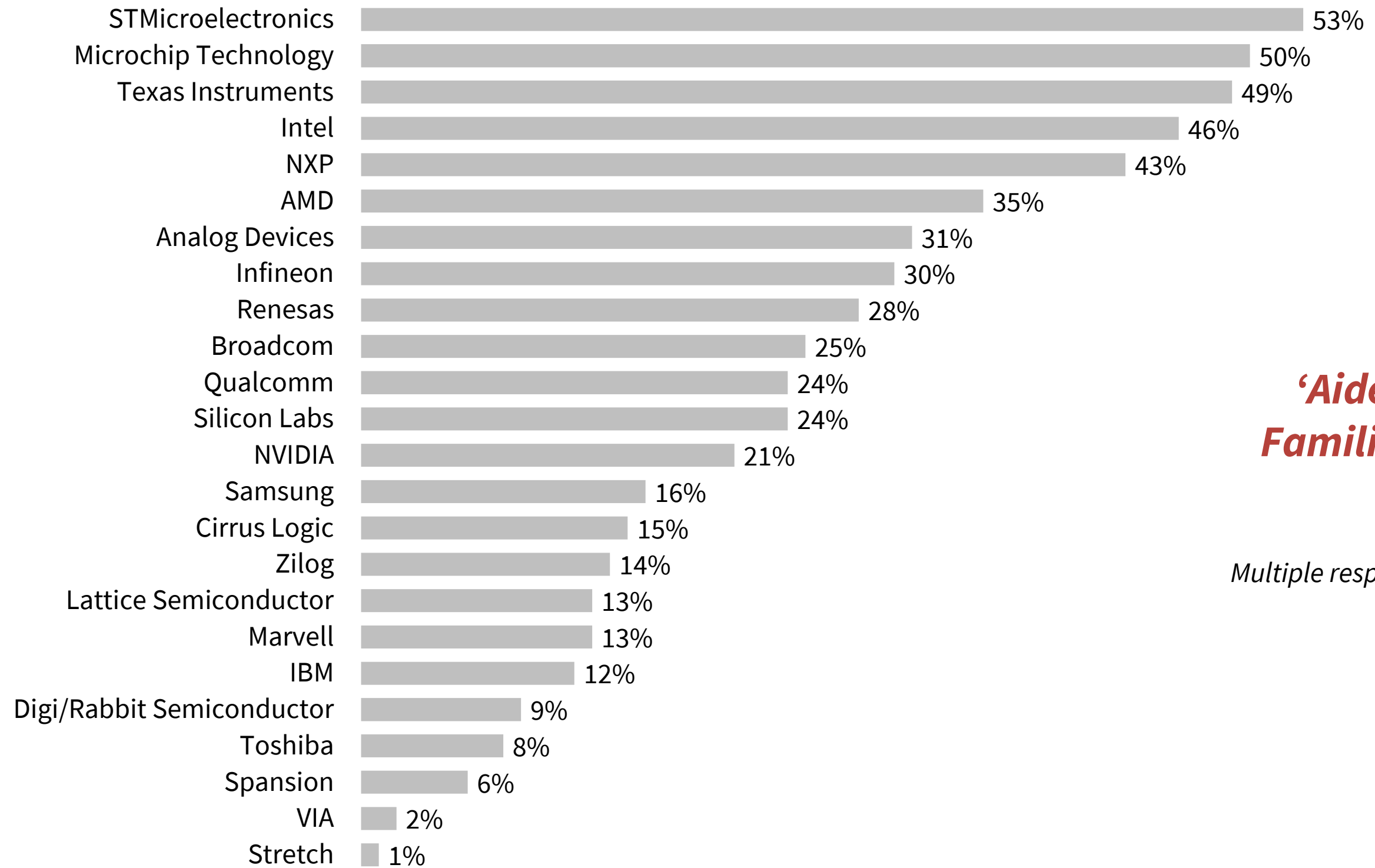
Factor	Percentage
The chip's performance	85%
The on-chip I/O or peripherals	85%
The software-development tools available for it (compilers, debuggers, etc.)	85%
The processor's debug support	81%
The "ecosystem" surrounding the chip (software, tools, RTOS, support, etc.)	79%
The software already available for it (middleware, drivers, existing code, etc)	79%
The hardware-development tools available for it (emulators, scopes, dev kits,…	78%
The chip's cost	77%
My familiarity with the architecture / chip family	74%
The supplier's reputation	73%
The chip family's future roadmap or growth path	70%
The chip's power consumption	70%
The operating systems it supports	69%
The chip's supplier/vendor	68%
The chip's security features	66%
The chip's functional safety features	61%
The chip's floating-point performance	58%
The chip's popularity	54%
Availability of on-chip programmable logic	53%
The chip's MAC/DSP features	52%
Open instruction set architecture	50%
The chip has an integrated RF or radio capability, (e.g., a "wireless MCU")	50%
The supplier's other product line(s)	50%
The processor's availability as IP (for ASIC/SoC development)	45%
The chip has vector (SIMD) arithmetic	41%
Virtual prototype/transaction-level model available for it (e.g., SystemC-based…	40%
The ability to customize or add instructions to the processor	39%

'Very Important' Summary

Total Respondents

Familiarity with MPU/MCU vendors

STMicro, Microchip, TI, Intel, and NXP are the most well-known processor vendors

Vendor	Percentage
STMicroelectronics	53%
Microchip Technology	50%
Texas Instruments	49%
Intel	46%
NXP	43%
AMD	35%
Analog Devices	31%
Infineon	30%
Renesas	28%
Broadcom	25%
Qualcomm	24%
Silicon Labs	24%
NVIDIA	21%
Samsung	16%
Cirrus Logic	15%
Zilog	14%
Lattice Semiconductor	13%
Marvell	13%
IBM	12%
Digi/Rabbit Semiconductor	9%
Toshiba	8%
Spansion	6%
VIA	2%
Stretch	1%

'Aided'
Familiarity

Multiple responses allowed

Total Respondents

Future consideration of MPU/MCU vendors

STMicro, NXP, TI, and Microchip are the most efficient at converting familiarity into consideration for their processor solutions

Vendor	Percentage
STMicroelectronics	68%
NXP	53%
Texas Instruments	51%
Microchip Technology	47%
Infineon	41%
Silicon Labs	40%
NVIDIA	39%
Renesas	38%
Intel	36%
AMD	35%
Analog Devices	34%
Broadcom	32%
Qualcomm	31%
Marvell	28%
Cirrus Logic	27%
Digi/Rabbit Semiconductor	25%
Lattice Semiconductor	22%
IBM	21%
Samsung	19%
Spansion	16%
Toshiba	11%
Zilog	10%
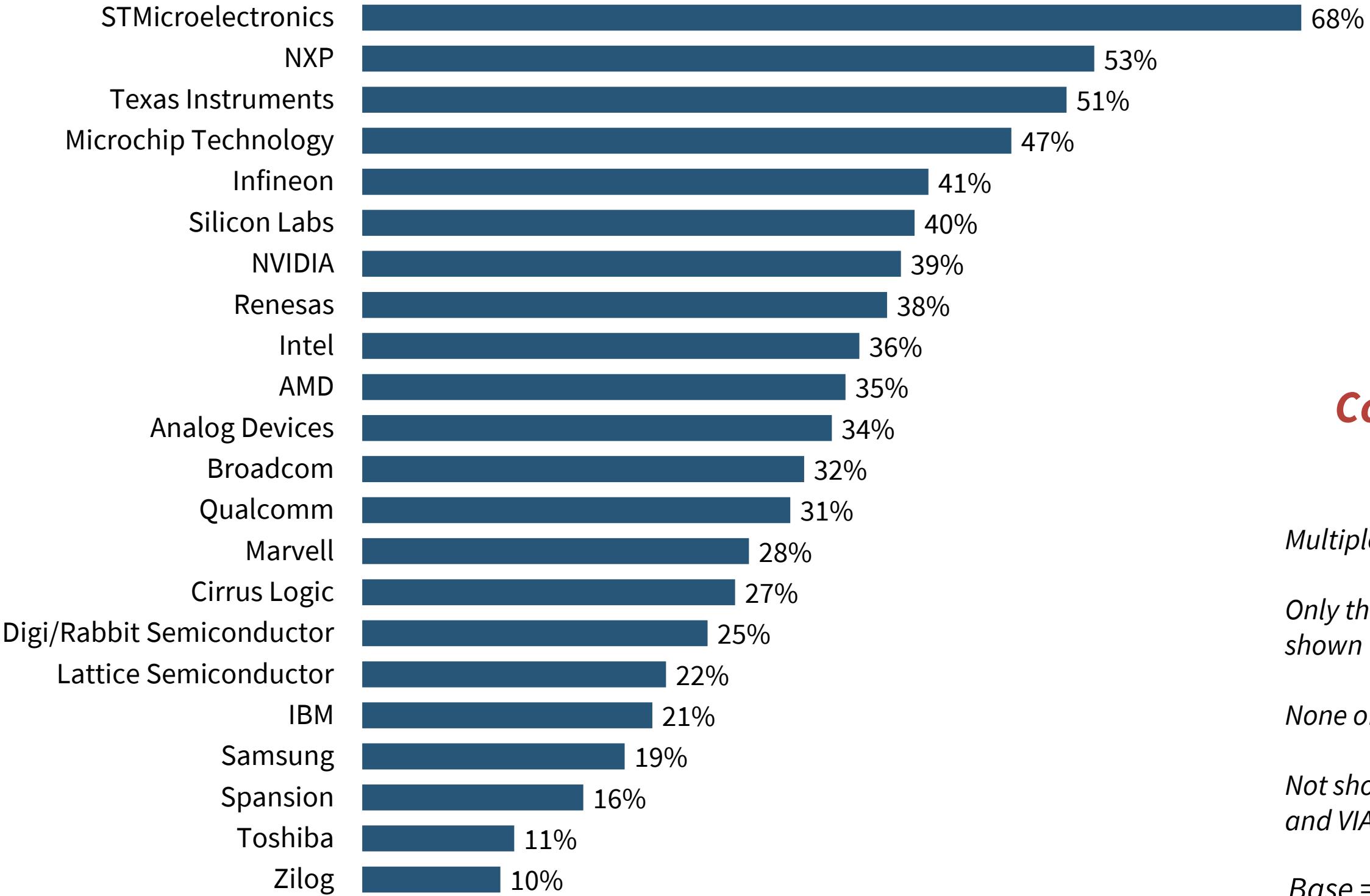
'Aided' Consideration

Multiple responses allowed

Only those with 4% or more total mentions shown

None of the above = 5%

Not shown due to small sample size: Stretch and VIA

Base = Those familiar with _each vendor_

Future consideration of 32-bit processor families

STMicro's STM32 is most widely considered, followed by Raspberry Pi, NXP's i.MX, Arduino and Microchip's AVR32

STMicro STM32 (ARM)	34%
Raspberry Pi	18%
NXP i.MX (ARM)	15%
Arduino	14%
Microchip (AVR32)	13%
Microchip PIC 32-bit (MIPS)	13%
AMD Zynq (with dual ARM Cortex-A9)	12%
Microchip SAMxx (ARM)	11%
Broadcom	9%
Microchip AT91xx/ATSAMxx (ARM)	9%
NXP Kinetis (ARM/Cortex-M4/M0)	9%
NXP LPC (ARM)	9%
AMD MicroBlaze (soft-core)	8%
TI Sitara (ARM)	8%
AMD Virtex-4 (with PowerPC 405)	7%
Intel Atom, Pentium, Celeron, Core 2, Core iX	7%
AMCC PowerPC 4xx	6%
AMD Fusion, Athlon, Sempron, Turion, Opteron	6%
AMD Virtex-5 (with PowerPC 405)	6%
Infineon PSOC 4 (ARM Cortex-M0) / PSoC 5 (ARM Cortec-M3)	6%
Intel SoC-FPGA (with dual ARM Cortex-A9)	6%
Renesas RX	6%
TI C2000 MCUs	6%
TI TM4Cx (ARM)	6%

Multiple responses allowed

Only those with 6% or more total mentions shown

Other = 7%
None of the above = 8%

Total Respondents

Future consideration of 16-bit processor families

Microchip's PIC24 / dsPIC, TI's MPS430 and STMicro's ST9 and ST10 are the 16-bit processors under consideration

Processor Family	Percentage
Microchip PIC24 / dsPIC	24%
TI MSP430	18%
STMicro ST9, ST10	17%
NXP HC16	12%
Intel 8086, '186, '286	11%
AMD 186, '188	10%
Renesas RL78	8%
Infineon XE166, XC2000, XC166, C166	8%
NXP HC12	8%
Renesas R8C	7%
Zilog Z180, Z380	3%

Multiple responses allowed

Other = 2%
None of the above = 36%
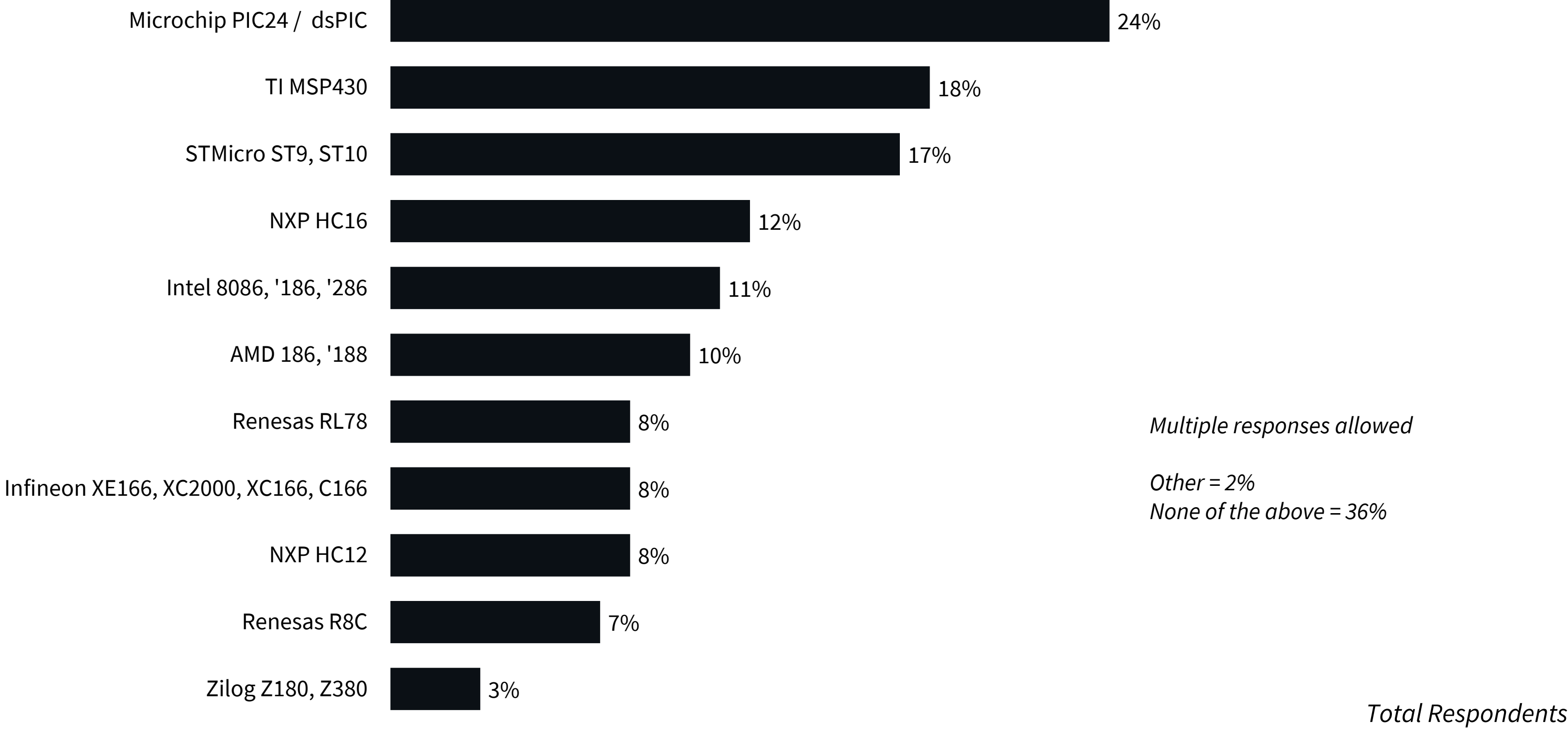
Total Respondents

Future consideration of 8-bit processor families

Microchip's PIC and AVR, STMicro's ST6, ST7 and ST8 and Intel's 80xx 8-bit processors are the most popular

Processor family	%
Microchip PIC	23%
Microchip AVR	20%
STMicro ST6, ST7, ST8	18%
Intel 80xx, '251	15%
Microchip 80xx	10%
AMD PicoBlaze (soft core)	9%
Infineon PSoC 1 (M8C) / PSoC 3 (8051)	9%
NXP / Philips P80x, P87x, P89x	8%
TI TMS370, 7000	8%
Infineon XC800, C500	7%
Renesas H8	5%
SiLabs 80xx	5%
Digi /Rabbit 2000, 3000	4%
NXP HC**	4%
Zilog Z8, Z80, Z180, eZ80	4%
Parallax	2%
Toshiba	1%
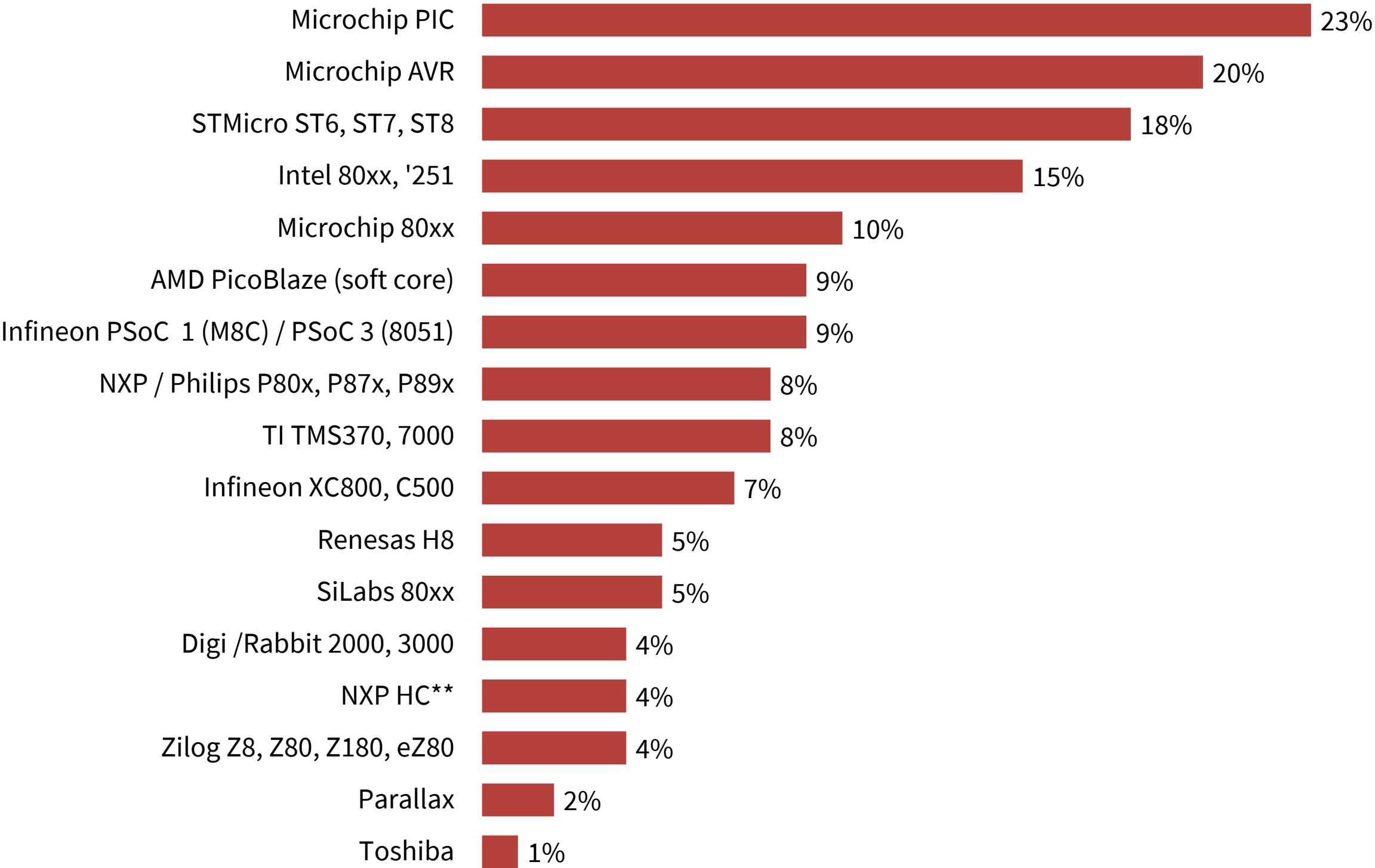
Multiple responses allowed

Other = 5%
None of the above = 31%

Total Respondents

Future consideration of DSP families

Microchip's dsPIC and ADI's ADSP-21xx, SHARC and TigerSHARC are the leaders in this category

44% would _not consider_ a DSP chip

Family	Percentage
Microchip dsPIC	22%
Analog Devices ADSP-21xx	13%
Analog Devices SHARC	11%
Analog Devices TigerSHARC	10%
NXP Cortex M4 LPC4000	10%
Analog Devices Blackfin	9%
NXP 563xx, 566xx, 568xx, 96xxx	9%
TI 'C6000	8%
TI 'C5000	7%
TI DaVinci	7%
TI KeyStone DSPs	4%
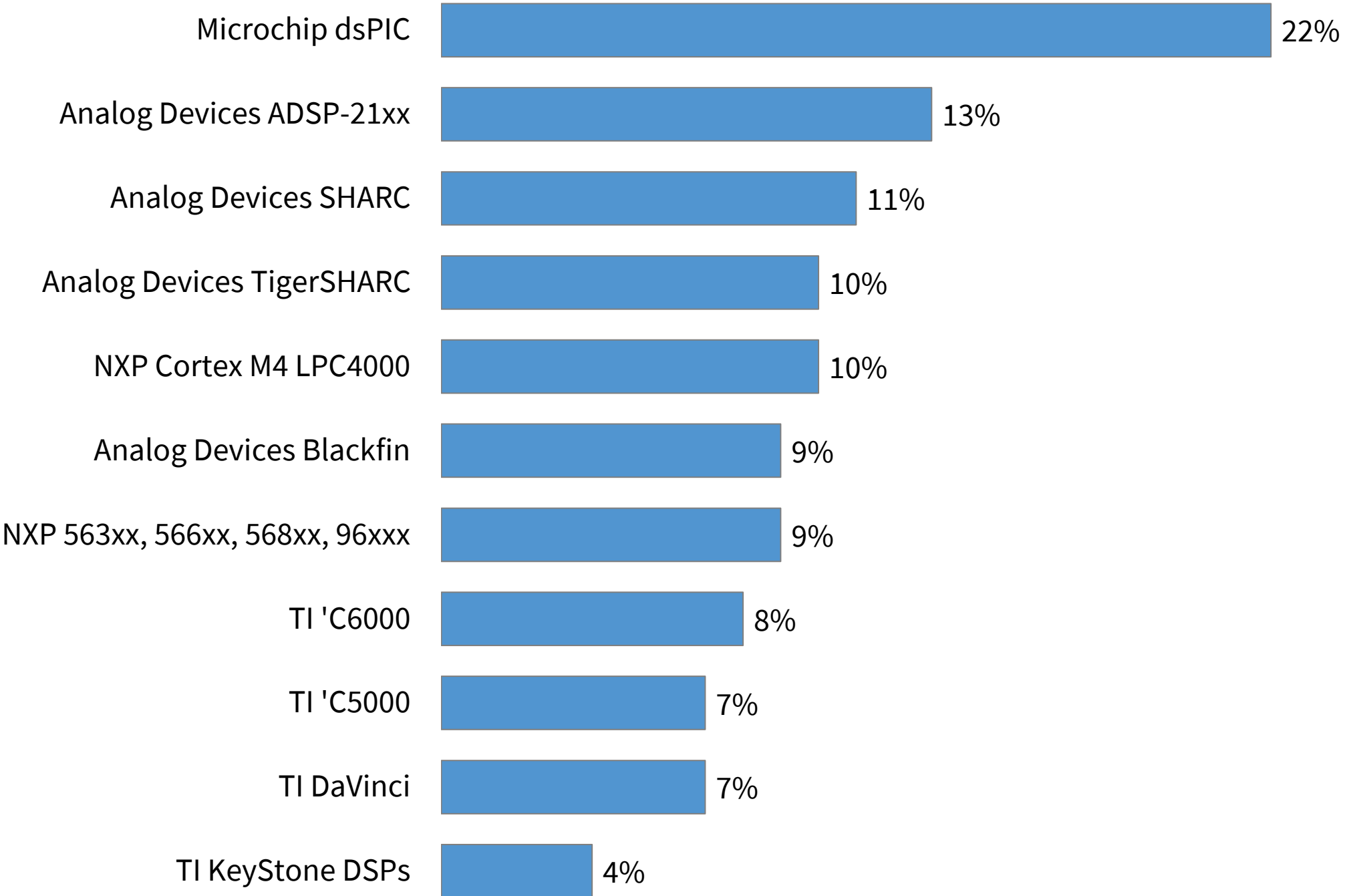
Multiple responses allowed

Other = 2%
None of the above = 44%

Total Respondents

Future consideration of FPGAs

AMD and Intel PSG are the most widely used vendors in the programmable logic space

*37% have incorporated FPGA chips in **current project***

Vendor	Percentage
AMD	35%
Intel PSG	27%
Lattice Semiconductor	22%
Cadence	17%
Microchip	17%
Efinix	12%
Achronix	11%
Siemens EDA (formerly Mentor Graphics)	11%
Gowin Semiconductor	9%
Flex Logic	9%
Synopsys/Synplicity	7%
Quicklogic	3%
Menta	1%

Multiple responses allowed

Other = 7%

Total Respondents

56. Does your current embedded project incorporate an FPGA chip?
57. Which of the following FPGA vendors and FPGA tool vendors do your current embedded projects use or consider using in the next 12 months?

ASPENCORE | 35

Design Tools

Software testing, simulation and emulation will grow in importance

EMEA and APAC teams are especially keen on SW test, simulation and emulation, and GUI system design

Category	Value
Software test	44%
Simulation	37%
Emulation	31%
Graphical system design	27%
Virtual prototyping	24%
Modeling in a high level language	22%
Cloud integration tools	22%
DevOps	22%
Application security test	20%

Multiple responses allowed
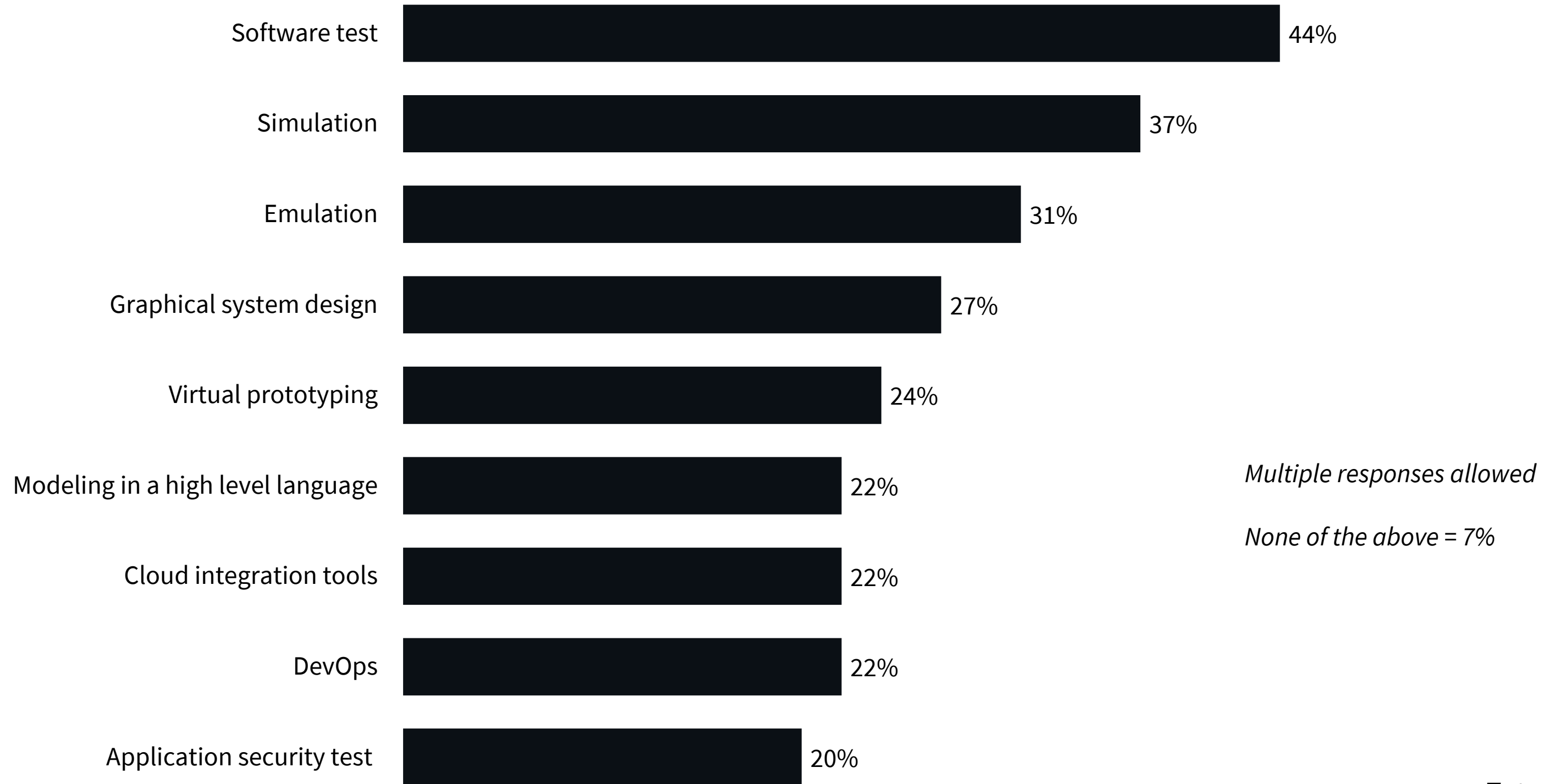
None of the above = 7%

Total Respondents

MATLAB is the most widely used system-level design tool

Also commonly used are FPGA prototypes (in the Americas) and System C language (in APAC and EMEA)

Tool	Percentage
MATLAB	32%
FPGA-based prototypes	22%
System C or other "hardware C" language	22%
LabVIEW	19%
Simulink	17%
UML	12%
QEMU	11%
Cadence Virtual System Platform	9%
Siemens (Mentor) Embedded Vista	8%
Synopsys Virtualizer	4%
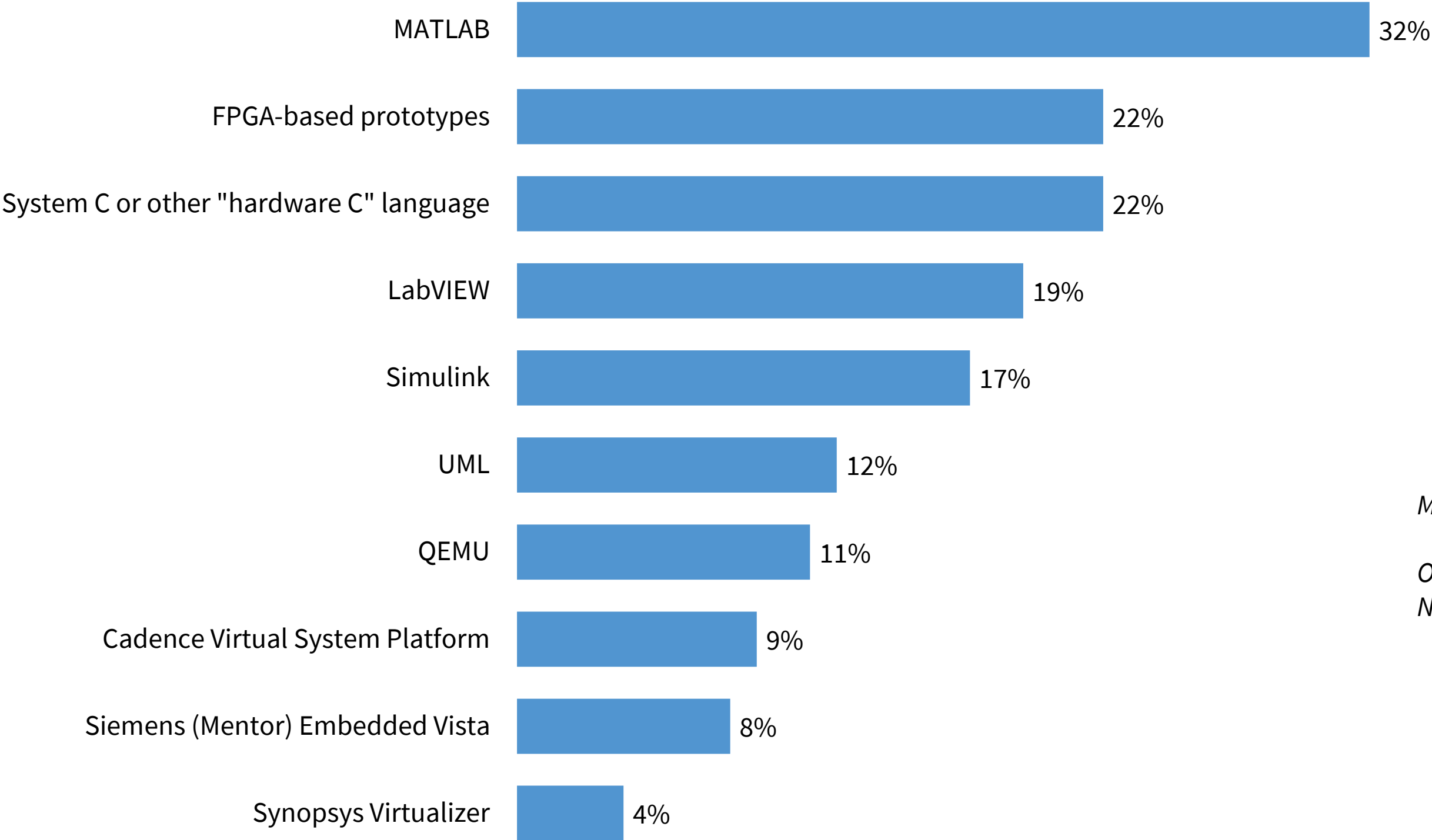
Multiple responses allowed

Other = 2%
None of the above = 26%

Total Respondents

embedded survey

Cloud integration tools are used for firmware updates and security management

More popular in the Americas, security and device management tools are more apt to be used in "new" designs

Firmware update — 35%

Security management (including auth and commissioning) — 26%

Device management (including device shadows) — 24%

Multiple responses allowed

Other = 1%
None / Don't use cloud integration tools = 43%

Total Respondents

63. Which cloud integration tools do you or your organization <u>currently use</u>?

ASPENCORE | 39

embedded survey
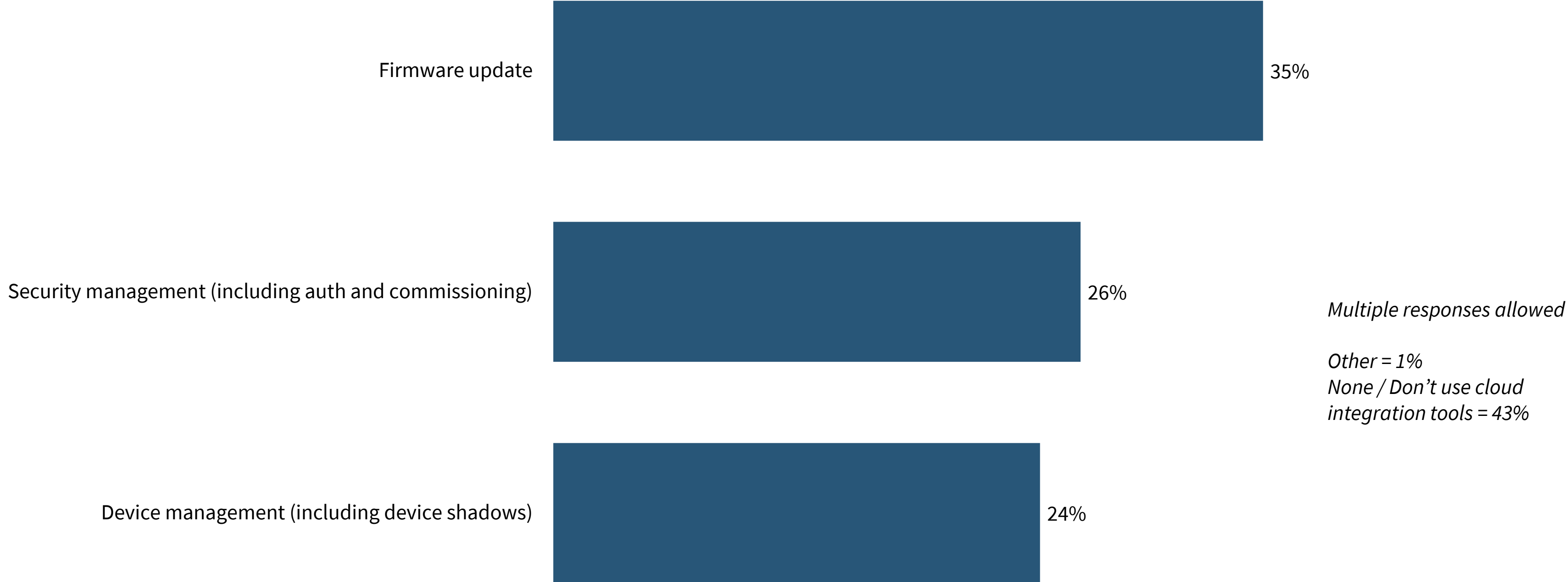
Most used project management platforms: MS Project and Excel

Managers more apt to depend on MS Project, IBM Telelogic DOORS, TeamCenter or open-source packages

Microsoft Excel — 36%
Microsoft Project — 33%
Open source tools — 31%
Visio — 18%
IBM Telelogic DOORS — 14%
Simulink — 9%
TeamCenter — 7%
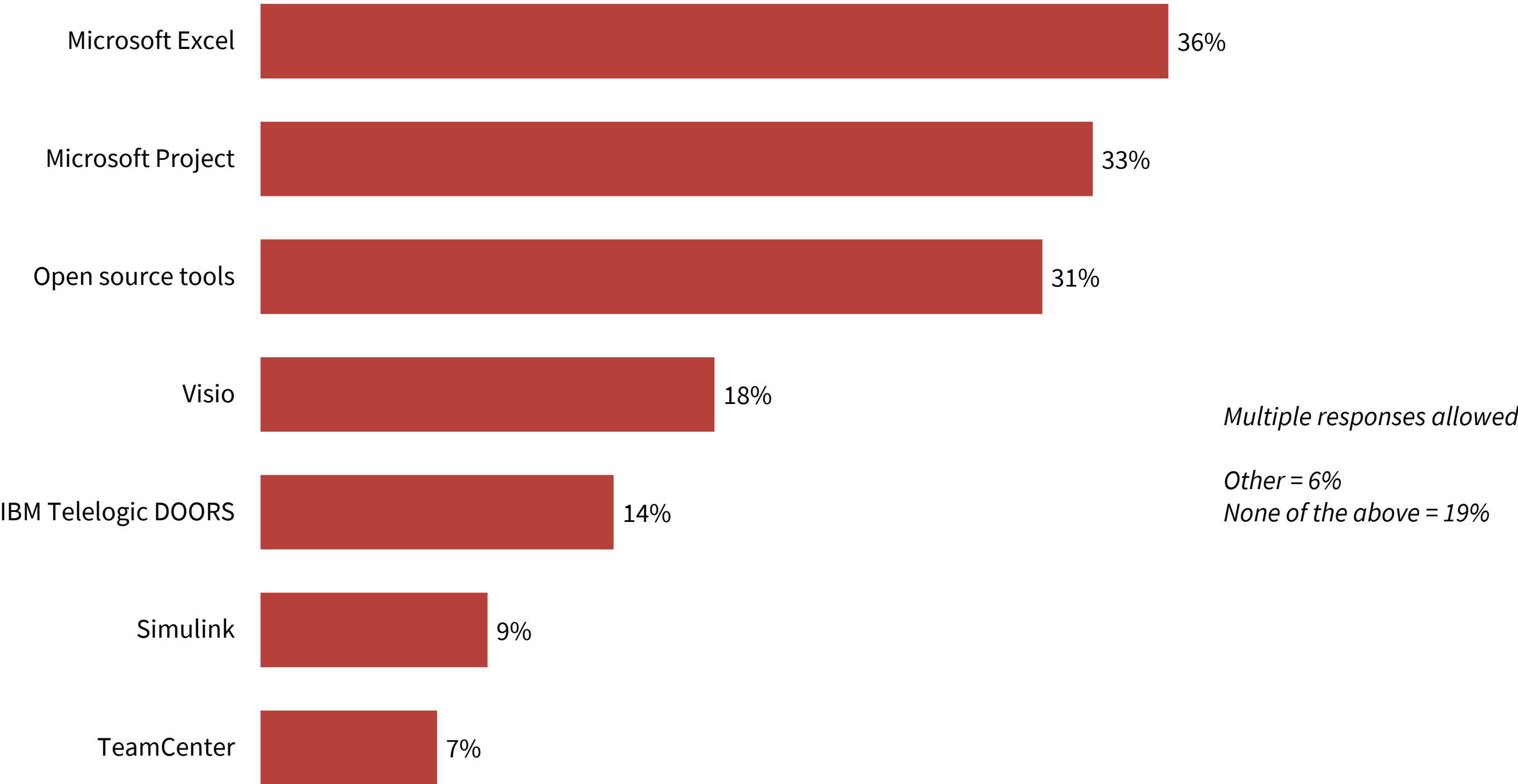
Multiple responses allowed

Other = 6%
None of the above = 19%

Total Respondents

64. Which of the following Project Management software packages do you currently use?

ASPENCORE | 40

Git is the most widely used version control software

Git and Subversion are especially popular in EMEA

Git — 63%

Subversion — 17%

CVS — 14%

Clearcase — 10%

Perforce — 9%

Multiple responses allowed

Other = 3%
None of the above = 14%
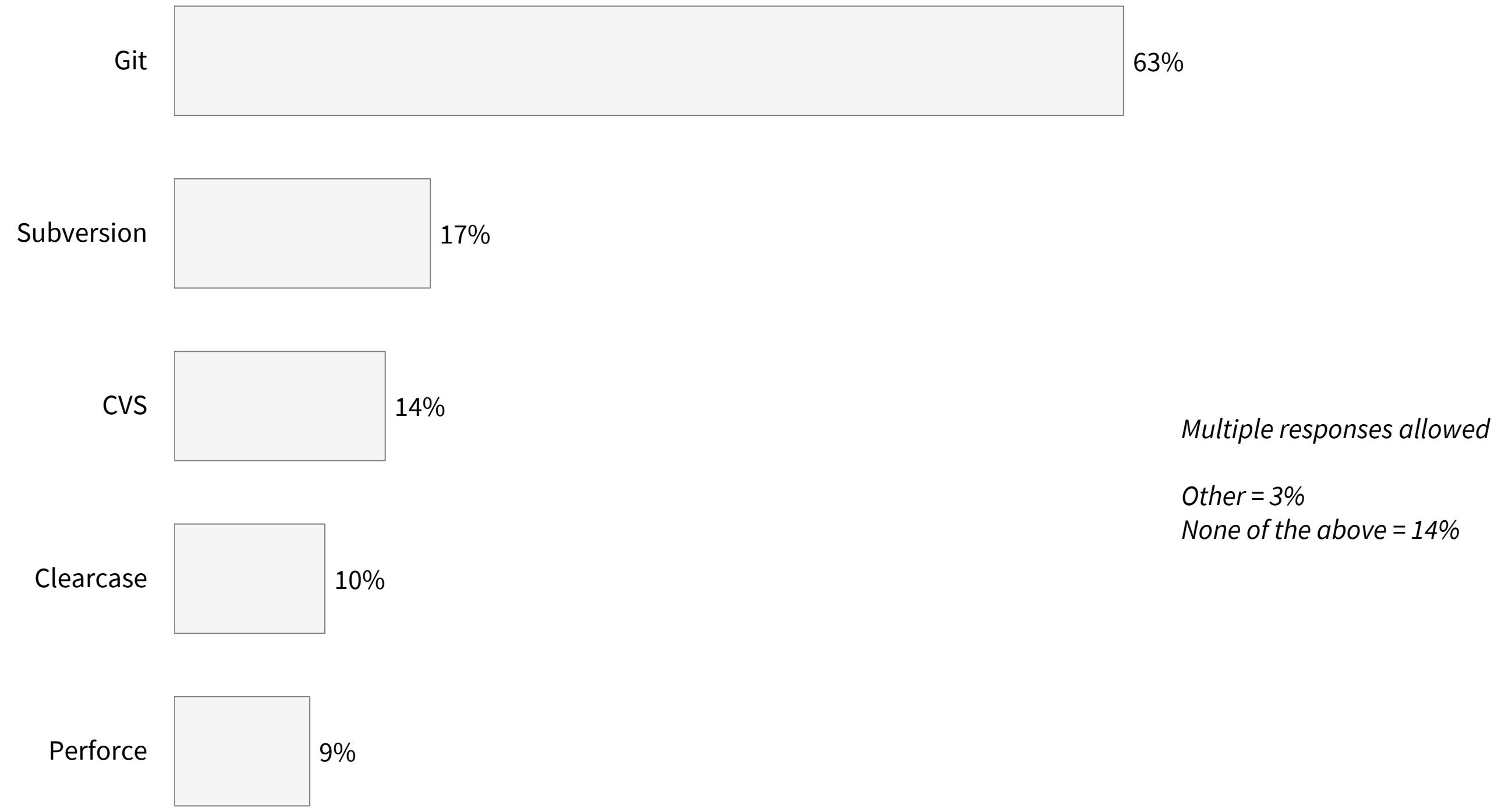
Total Respondents

Information, Training & Continuing Education

Most utilized methods for self-education on embedded development trends – online training, plus vendor white papers and professional/technical journals

Early career engineers are more apt than older peers to also consider training provided from professional associations and private companies

Most effective ways to maintain professional skills

Method	Percentage
Online training/webinars provided by vendors	40%
Technical white papers from vendors	36%
Professional/technical journals	36%
Online training/webinars provided by distributors	24%
Online training/webinars provided by media organizations (like EE…	20%
Conferences-seminars provided by vendors	20%
Online training/webinars provided by professional associations (like…	19%
Professional development courses from private companies	18%
On-site seminars given by vendors	17%
Professional development courses from universities offered online	15%
Certification training	14%
Conferences-seminars provided by professional associations	13%
Professional development courses from university extension programs	11%
Conferences-seminars provided by media organizations (like ESC)	8%
Books	5%

Total Respondents

Embedded engineers devote two full work weeks to formal training

APAC and earlier career engineers spend 20% more time in training than do their colleagues

Mean = 10.2 days per year

Category	%
15 days or more	21%
6 to 14 days	31%
4 to 5 days	18%
3 days or less	19%
No days	11%
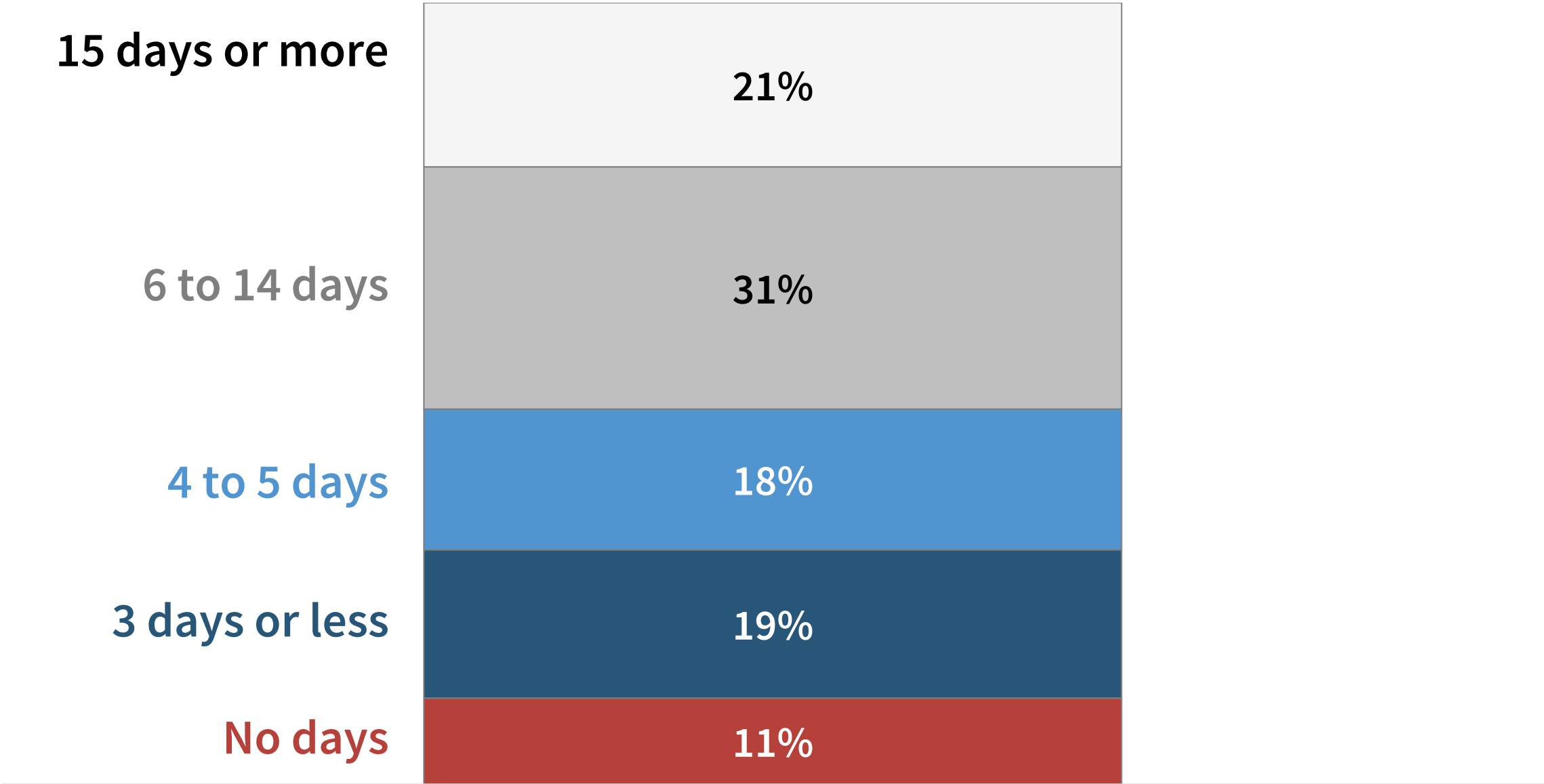
Select up to FOUR

Other = 1%
I do not have a systematic/formal method of advancing my professional skills = 4%

Days per year spent on formal career-based training

Total Respondents

Observations

EMBEDDED.COM

The Embedded Development Environment

- **Most embedded projects involve incremental upgrades to existing designs (such as adding software features or new MPUs/MCUs)**
  - Most projects are developed for industrial automation and instrumentation, IoT, communications, and automotive
  - Current embedded development devotes considerable attention to performance, connectivity, power efficiency and signal processing

- **Embedded development teams (containing, on average, 20 engineers) have burgeoning workloads**
  - Teams typically tackled over 4 projects in the past year, with two in the development pipeline at any point in time
  - Embedded projects take 8 months on average to complete, with SW design requiring 50% more development time than hardware
  - Reuse of software code, hardware and IP is common, as is the use of development boards (particularly Raspberry PI and Arduino)

- **Most pressing embedded design challenges - meeting performance specs, choosing the right processor and test/debugging, along with safety, security and power management**
  - Over one-third of embedded designs incorporate wireless capabilities
  - IP theft, product tampering, and cloning are primary security issues, especially for larger OEMs
  - Nearly one-third of embedded design is devoted wholly or partially to IoT applications, most for sensor-driven, industrial or mobile communications
  - Embedded AI and machine learning attract considerable attention, followed by embedded vision and speech capabilities

Operating Systems

- **Most embedded projects will increasingly utilize an operating system**

    - Four in use either commercial OS or open-source OS distributed commercially, but nearly 30% of those now using commercial OS are considering open-source alternatives

- **Leading OS selection criteria – processor support, tools, and overall cost**

    - Also high on the agenda: security, design customizability and flexibility, and ease of use

- **Most popular embedded OSs – Embedded Linux, FreeRTOS, Ubuntu, Debian and Android**

- **Half use or plan to use embedded hypervisors**

    - Key reasons: support for safety and security, and separation of multiple real-time and legacy applications as well as "guest" operating systems

Microprocessors / Microcontrollers / FPGAs

- **Most embedded projects use multiple processors or multicore solutions**

  - Typical embedded designs include 2+ MPUs/MCUs (particularly in the Americas), although clock rates are faster among Asian developers.

  - 32-bit processors continue to be the most prevalent, and 39% upgraded to larger processors in the past year

- **Half of the embedded designs contain different MPUs/MCUs than were used previously in order to obtain access to more features and a clear roadmap for the future**

  - Among those using different processors, half chose from within the same family or architecture, while another half chose a different architecture or family

  - Among the important processor selection criteria are performance, available peripherals, HW/SW tool environment, and support ecosystem

- **STMicro, Microchip, TI, Intel, and NXP are the most well-known processor vendors**

  - STMicro's STM32 is the most widely considered 32-bit processor along with Raspberry Pi, NXP's i.MX, Arduino and Microchip's AVR32, while Microchip's PIC24 / dsPIC, TI's MPS430 and STMicro's ST9 and ST10 are the 16-bit processors most under consideration, and Microchip's PIC and AVR, STMicro's ST6, ST7 and ST8 and Intel's 80xx 8-bit processors are the most popular

  - Widely considered DSPs include Microchip's dsPIC and ADI's ADSP-21xx, SHARC and TigerSHARC

  - AMD and Intel PSG are the best-known and used vendors in the programmable logic space

Design Tools

- **Software testing, simulation and emulation will grow in importance as embedded design techniques**

  - EMEA and APAC teams are especially likely to utilize these approaches

- **Embedded designers utilize a wide variety of SW/HW design tools – including compilers, debuggers, oscilloscopes, logic analyzers, design environments, and SW libraries**

  - EMEA embedded development teams are more likely than are peers elsewhere to use these tools

- **MATLAB is the most widely used system-level design tool**

  - Also commonly used are FPGA prototypes (in the Americas) and System C language (in APAC and EMEA)

- **Cloud integration tools are used for firmware updates and security management**

  - More popular in the Americas, security and device management tools are more apt to be used in "new" designs

- **Most used project management platforms: MS Excel and MS Project**

- **Git is the most widely used version control software solution**

Information, Training & Continuing Education

- **Vendor websites (SW and HW), white papers, standards-related publications, and webinars/webcasts join search engines as the most widely used information sources**
    - Other than social media platforms, early career embedded developers trail their more experienced colleagues in their use of nearly all content delivery vehicles

- **Embedded developers eagerly consume vendor-supplied training resources**
    - The typical embedded engineer devotes roughly two work weeks per year to formal training
    - Early career engineers also look to professional associations and private 3rd parties for additional training
    - Popular industry events showcasing embedded development include Embedded Linux Conference (ELC), Embedded Systems Conference, CES and Embedded World

- **In addition to online training, widely utilized methods for self-education on trends in embedded development include vendor white papers and professional/technical journals**
    - Roughly one hour a day on average is allocated to reading technical publications

Questions and Answers

EMBEDDED.COM

Thank You

EMBEDDED.COM