# Surveying the Community:

## What Could Possibly Go Wrong?

ANASTASIA KAZAKOVA

Cppcon

The C++ Conference

2023

October 01 - 06

# About me

C++: Embedded, Telecom, 4G/LTE

@anastasiak2512

C++ Tools PMM

The Dutch C++ Group

# Goals

1. Share recent data sources about C++ ecosystem
2. Explain how to read them correctly
3. Learn popular misconceptions
4. Help you escape typical biases and errors

# Basic calculations

How many developers are there in the world?

How many professional developers are there in the world?

How many C++ developers are there in the world?

How many professional C++ developers are there in the world?

## Professional developers

Full-, self- or part-time employed devs and freelancers whose one of job roles is QA, Dev, Team Lead, DBA, Architect, DevOps or Technical support.

# How many professional developers are there in the world?

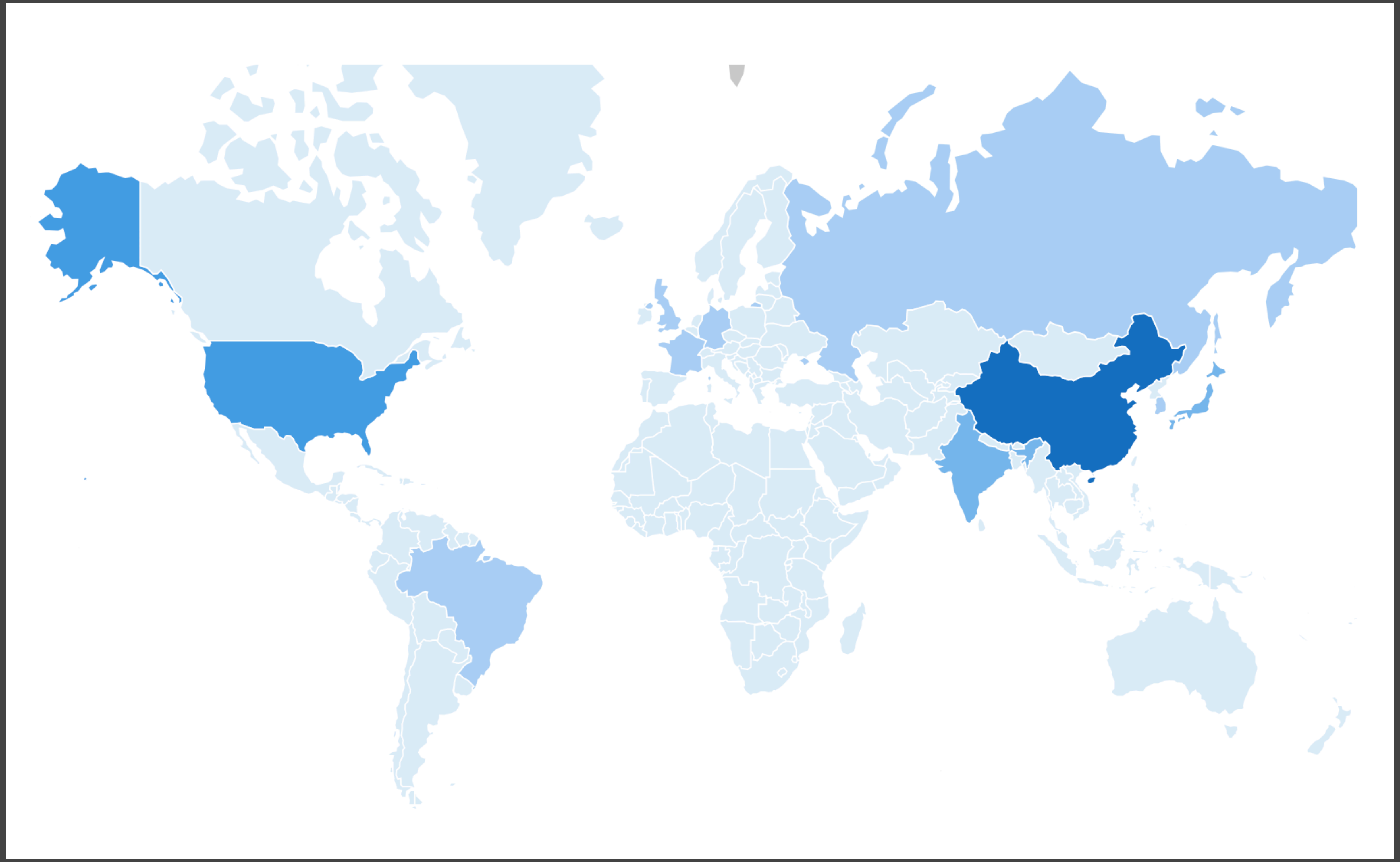
–

12,300,000 - 14,878,000

professional developers worldwide in 2022

# How many professional developers are there in the world?

1. China 2,643K
2. US 1,840K
3. Japan 1,342K
4. India 1,201K
5. Germany 496K
6. UK 401K
7. Brazil 389K

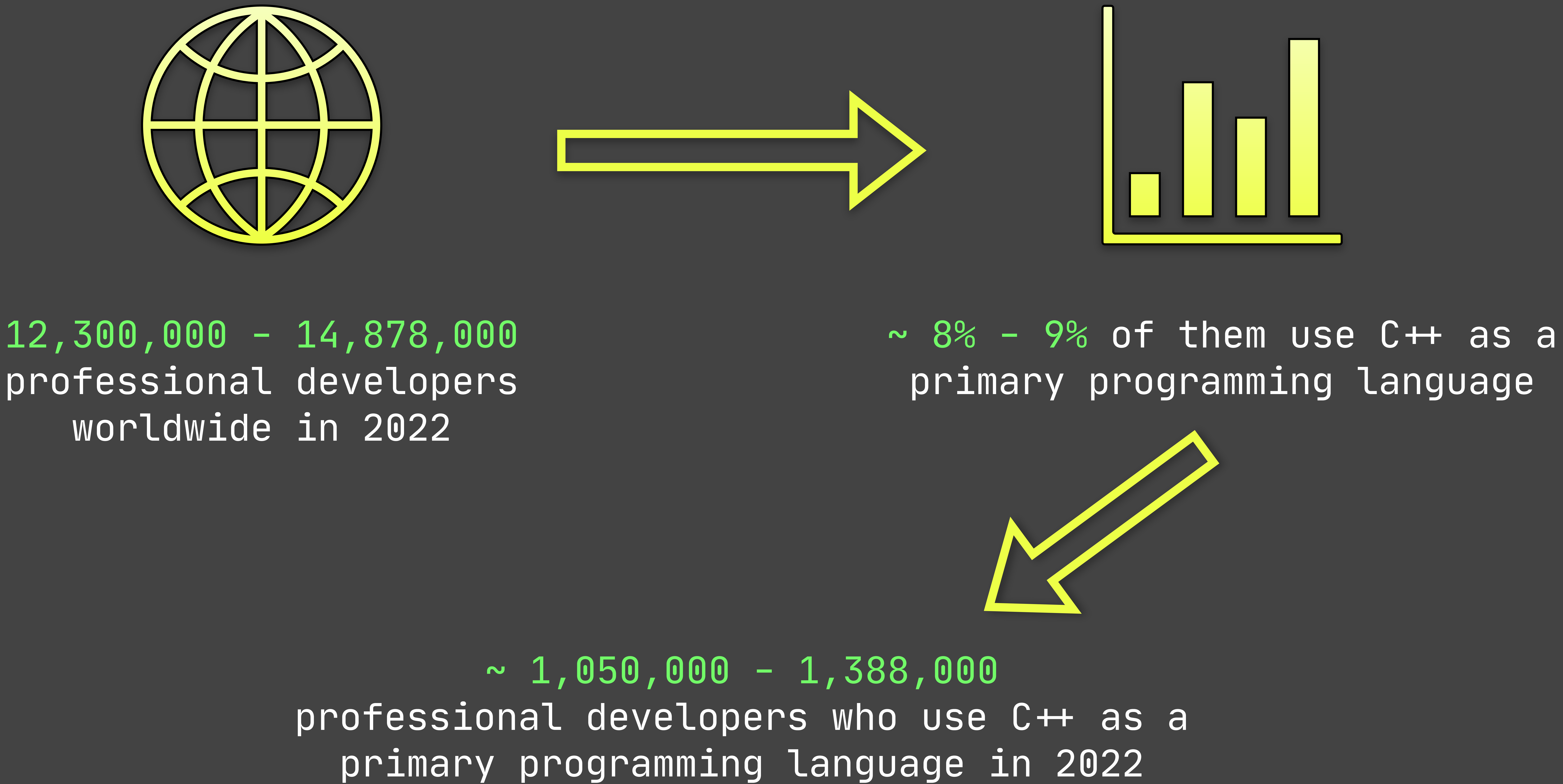


## C++ shares

-

- Developer Ecosystem Survey
- StackOverflow Survey

# C++ shares



12,300,000 - 14,878,000 professional developers worldwide in 2022

~ 8% - 9% of them use C++ as a primary programming language

~ 1,050,000 - 1,388,000 professional developers who use C++ as a primary programming language in 2022

# C++ shares

-

1, 157, 000

professional developers with C++ as primary language in 2023

2, 492, 000

professional developers who regularly use C++ in 2023

# C++ shares

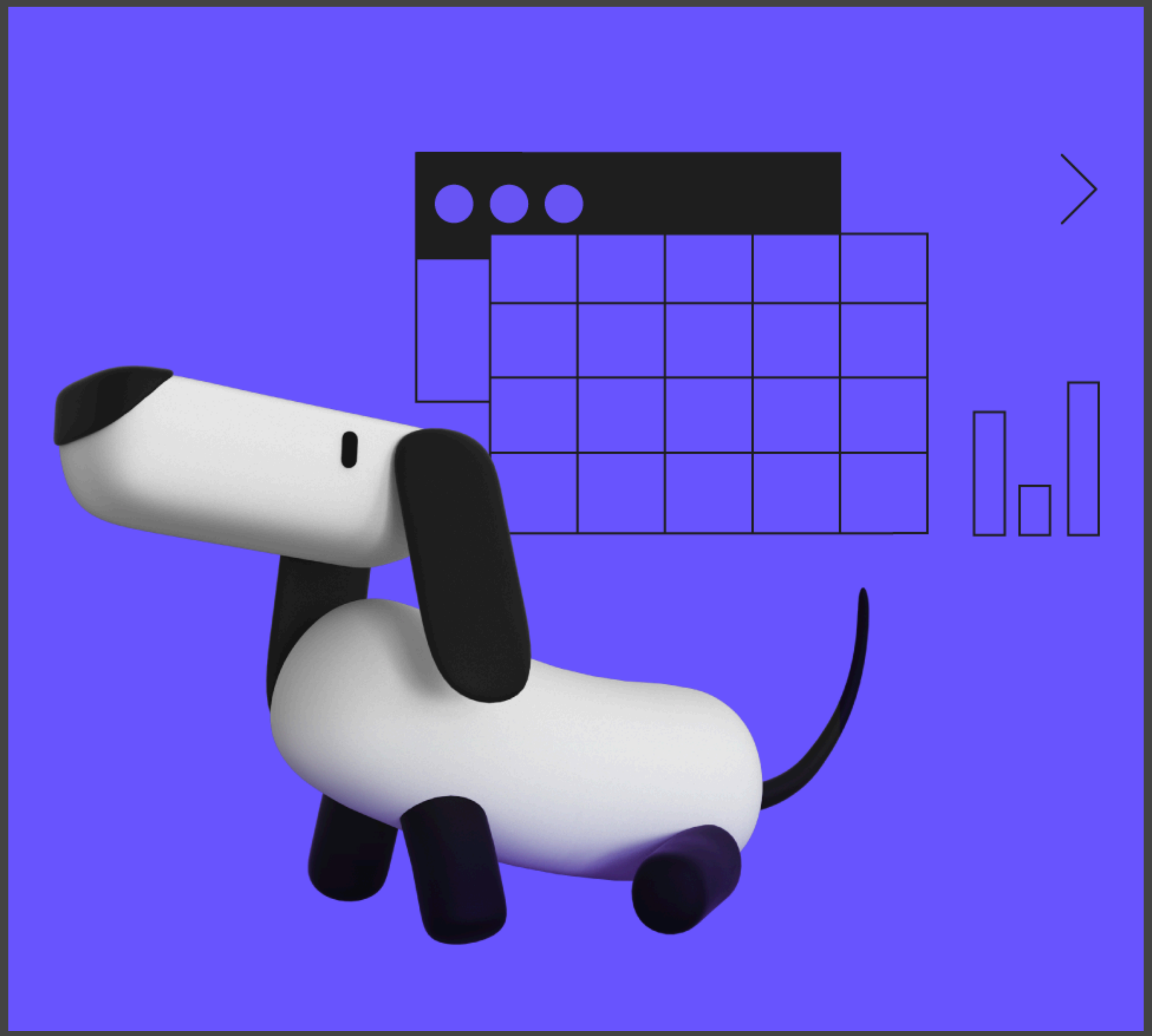
Developer Ecosystem language markets based on primary languages
2018
2019
2020
2021
2022
2023
Proportion / forecast
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
JavaScript
Java
Python
C#
SQL
C++
TypeScript
Go
C
Shell scripting languages
Kotlin
Ruby
Scala
Swift
Objective-C
Groovy
Visual Basic
Elixir
R
Rust
Dart

# C++ shares

Developer Ecosystem language markets based on used languages

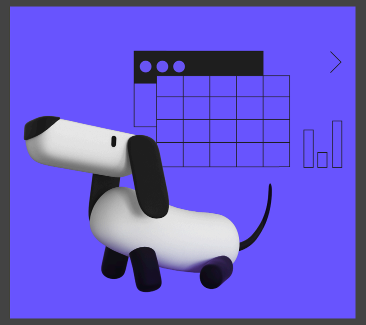
● 2019 ● 2020 ● 2021 ● 2022 ● 2023
Proportion / forecast
10,000,000
9,000,000
8,000,000
7,000,000
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
JavaScript
SQL
Java
Python
Shell scripting languages
TypeScript
C#
Go
C++
C
Swift
Ruby
Objective-C
Scala
Groovy
Dart
Visual Basic
Rust
R
Elixir
Kotlin

# Community Surveys in C++

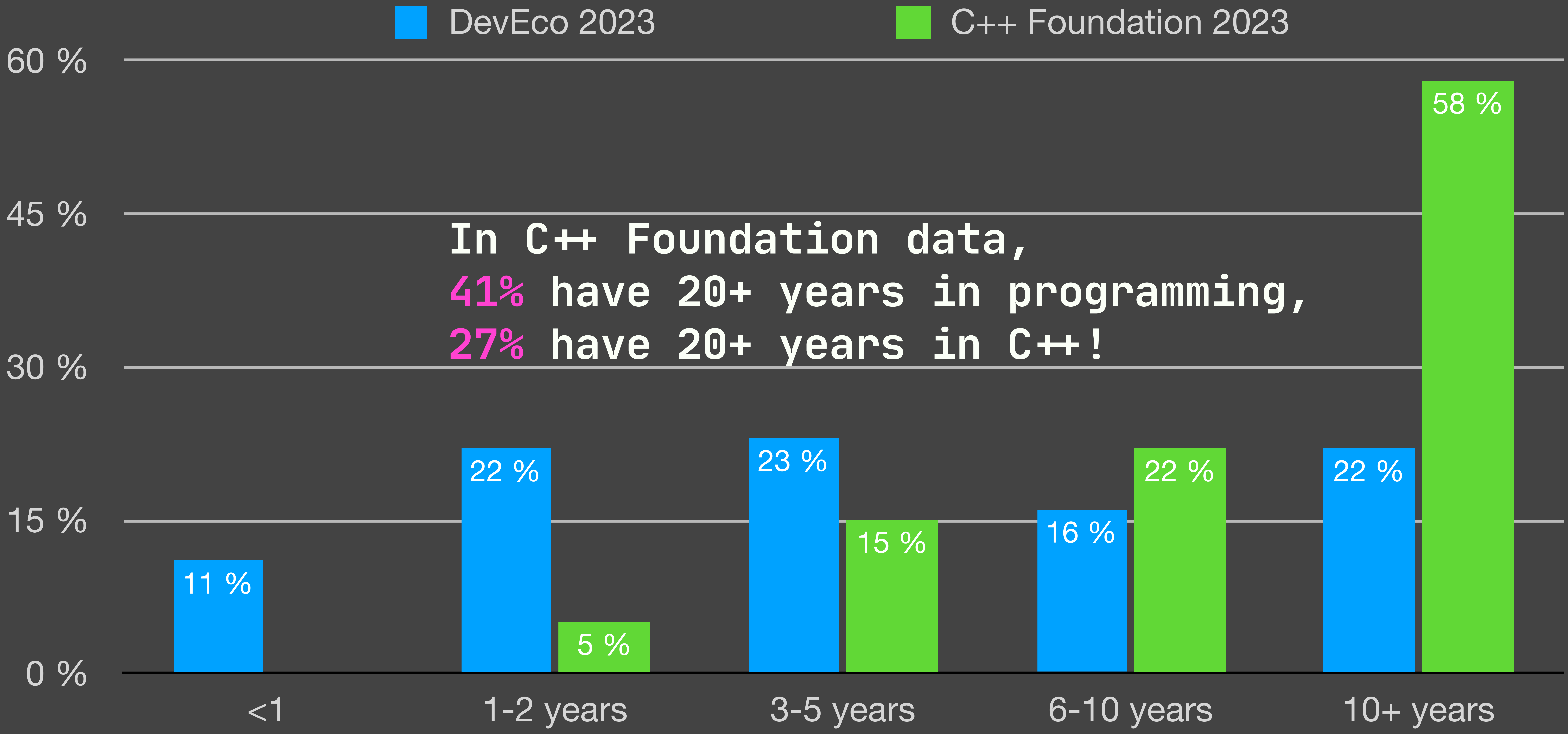


Meeting C++

# Community Surveys in C++

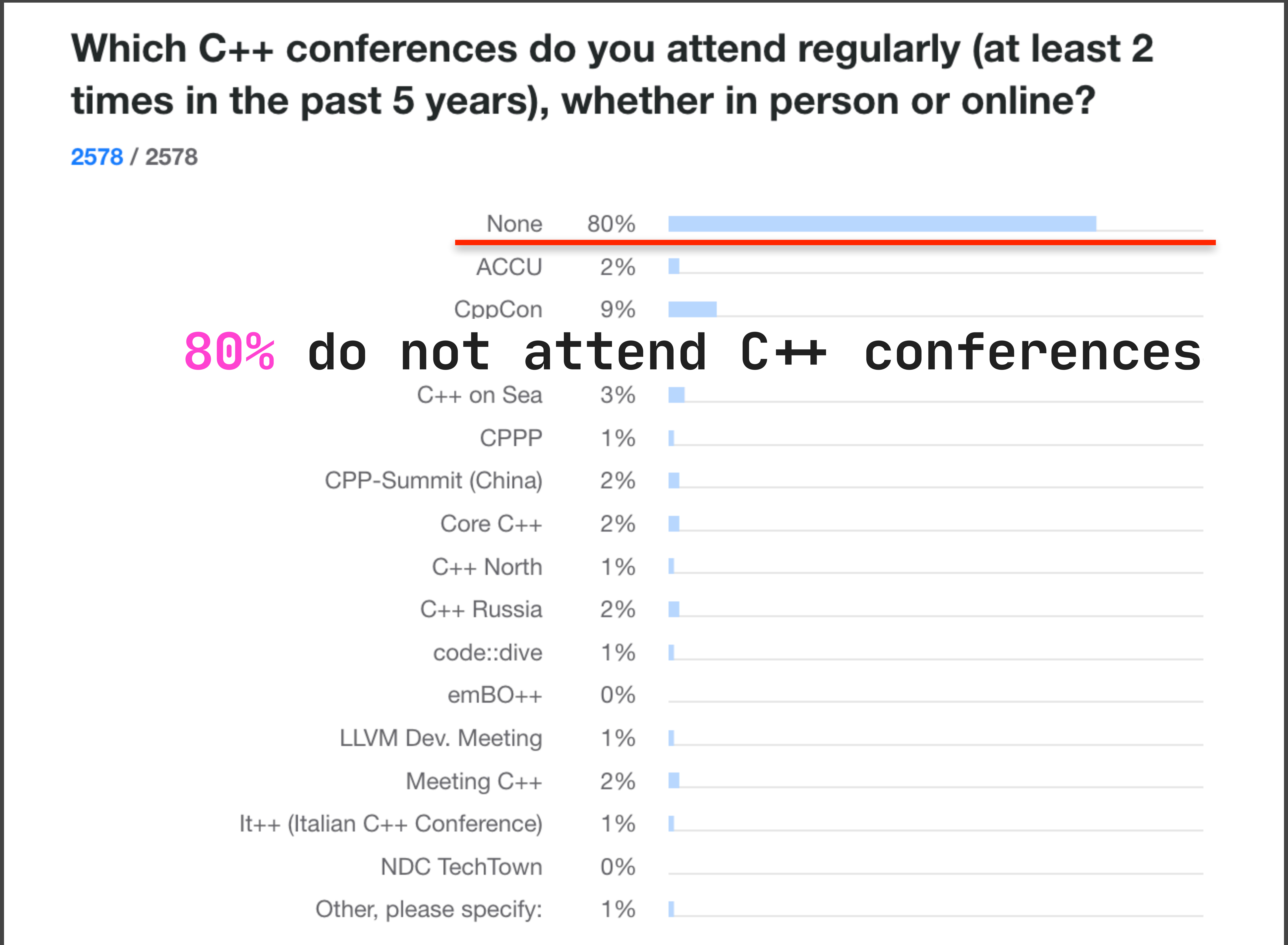
|  | C++ | Meeting C++ |
|---|---|---|
| Running since 2017 | Running since 2018 | Running since 2020 as a continuous survey |
| Global for all languages | Focused on C++ | Focused on C++ |
| Partially-random set from 527 questions | 21 questions | 88 questions |
| In 2023: 2,647 responses | In 2023: 1,722 responses | ~ 1,000-1,500 in 2020/21,<br>~ 3,000 through years |

# Audience



In C++ Foundation data,
41% have 20+ years in programming,
27% have 20+ years in C++!

# Audience



**Which C++ conferences do you attend regularly (at least 2 times in the past 5 years), whether in person or online?**

2578 / 2578

| | |
|---|---|
| None | 80% |
| ACCU | 2% |
| CppCon | 9% |
| C++ on Sea | 3% |
| CPPP | 1% |
| CPP-Summit (China) | 2% |
| Core C++ | 2% |
| C++ North | 1% |
| C++ Russia | 2% |
| code::dive | 1% |
| emBO++ | 0% |
| LLVM Dev. Meeting | 1% |
| Meeting C++ | 2% |
| It++ (Italian C++ Conference) | 1% |
| NDC TechTown | 0% |
| Other, please specify: | 1% |

80% do not attend C++ conferences

## Reducing the error
–

Every survey is lying

What? How? When? Where?

## Reducing the error

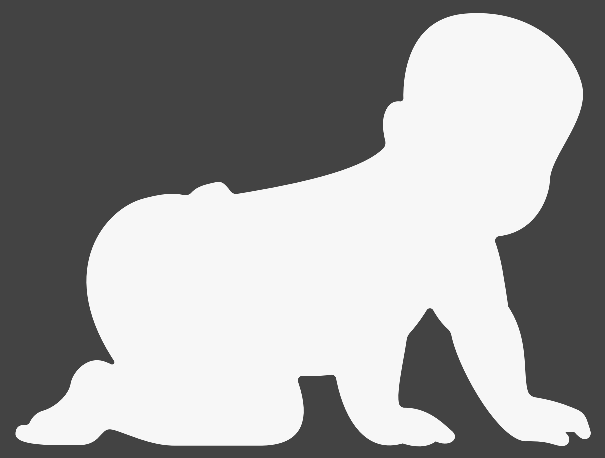
-

1. Sampling bias (data cleaning)
2. Response burden bias
3. Brand bias
4. Regional bias
5. Language (EN) bias
6. Statement bias

## Reducing the error: Sampling bias

–

1. No primary programming language
2. Filled-out too fast
3. Identical:
   - 75% identical answer from identical IPs
   - or same email address
4. Conflicting answers



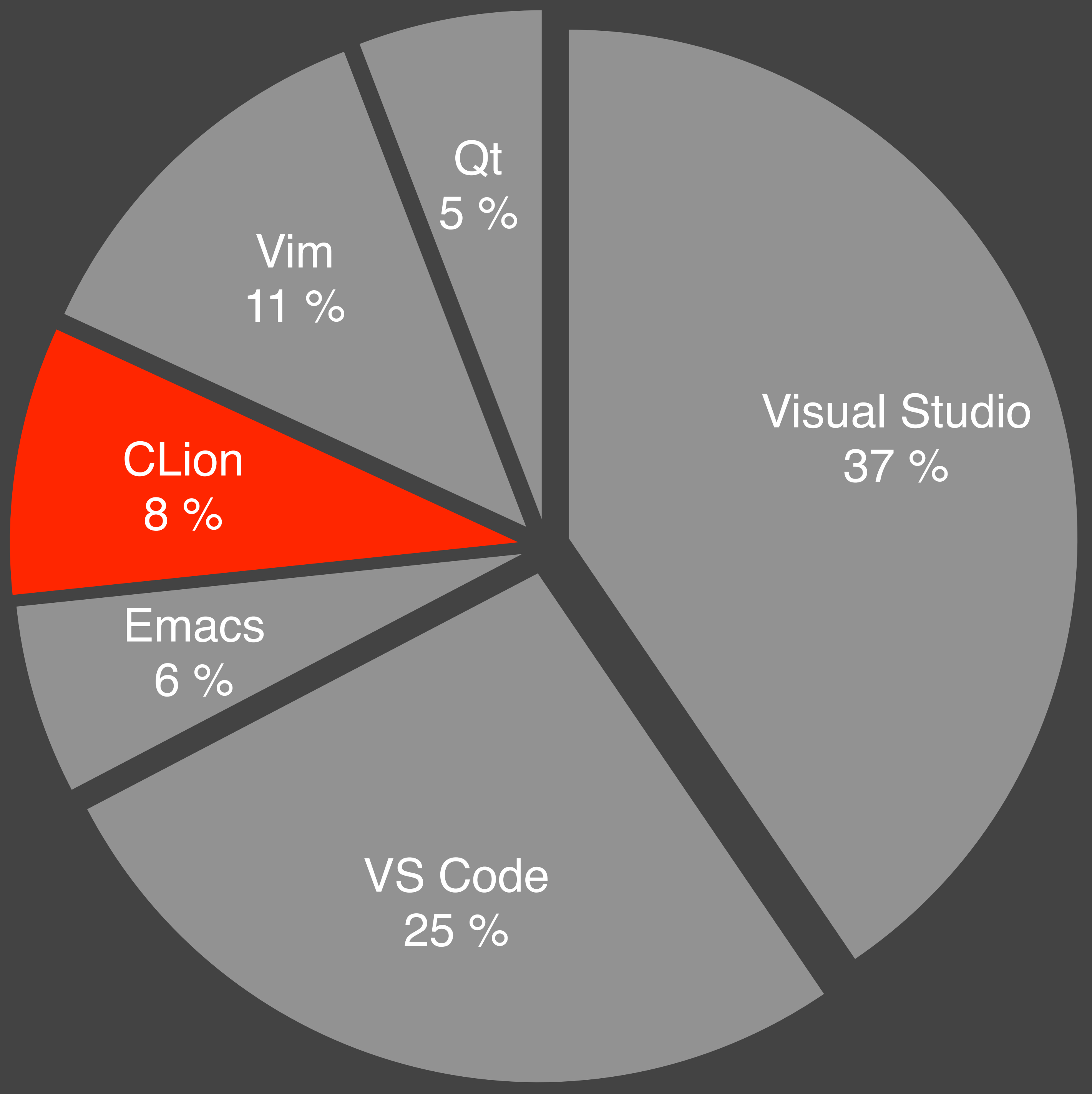
18-20yo and 16+ years of experience
18yo CEO with 10 years of experience

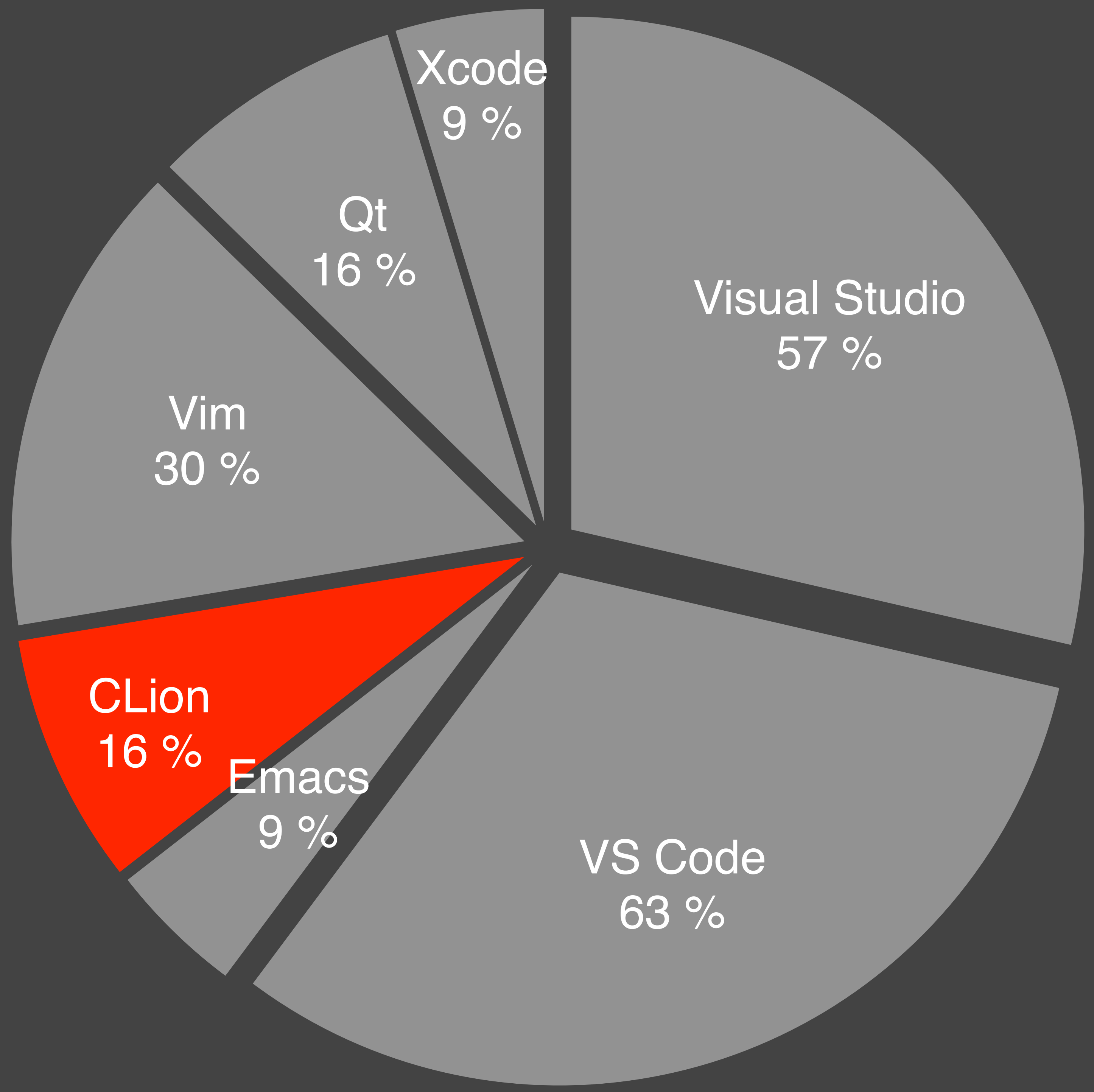
## Reducing the error: Responses burden

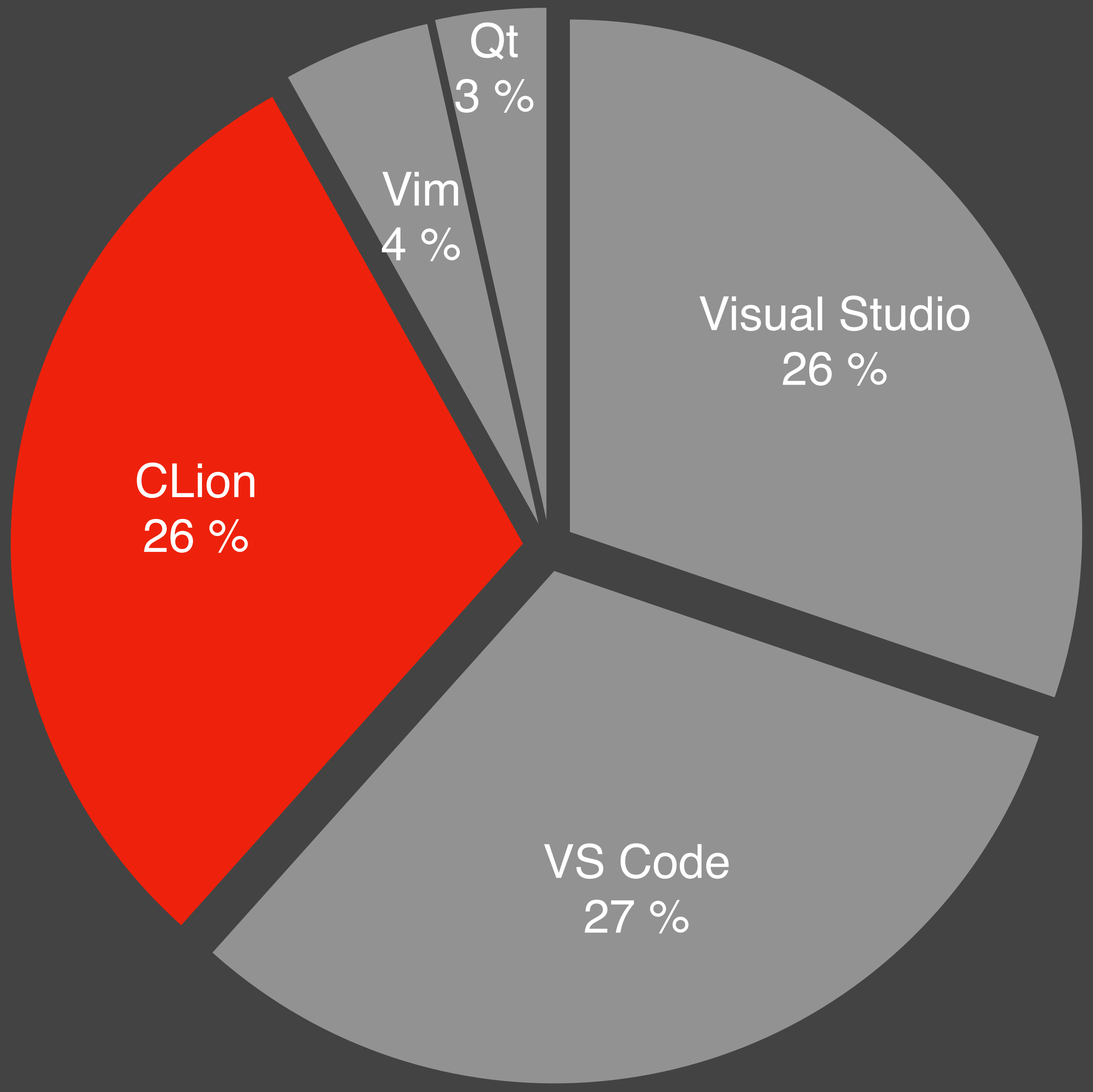
1. Total 527 questions
2. Primary languages in priority
3. Regularly used → randomly show to 50% of qualified respondents
4. Randomize 8 extra sections → show 2
5. 30-40 minutes

**The C++ Stories Weekly Newsletter**
**11,000 readers**
**243 full C++ answers in 2023!**

# Reducing the error: Brand bias


C++ Foundation Primary IDE 2023


C++ Foundation Total 2023


DevEco 2023 C++ IDE

C++ IDE: WebStorm 3, IntelliJ IDEA 37

# Reducing the error: Regional bias

–

17 countries contains the majority of developers

In 14 countries, at least 300 responses from external sources, such as ads

# Reducing the error: Language bias

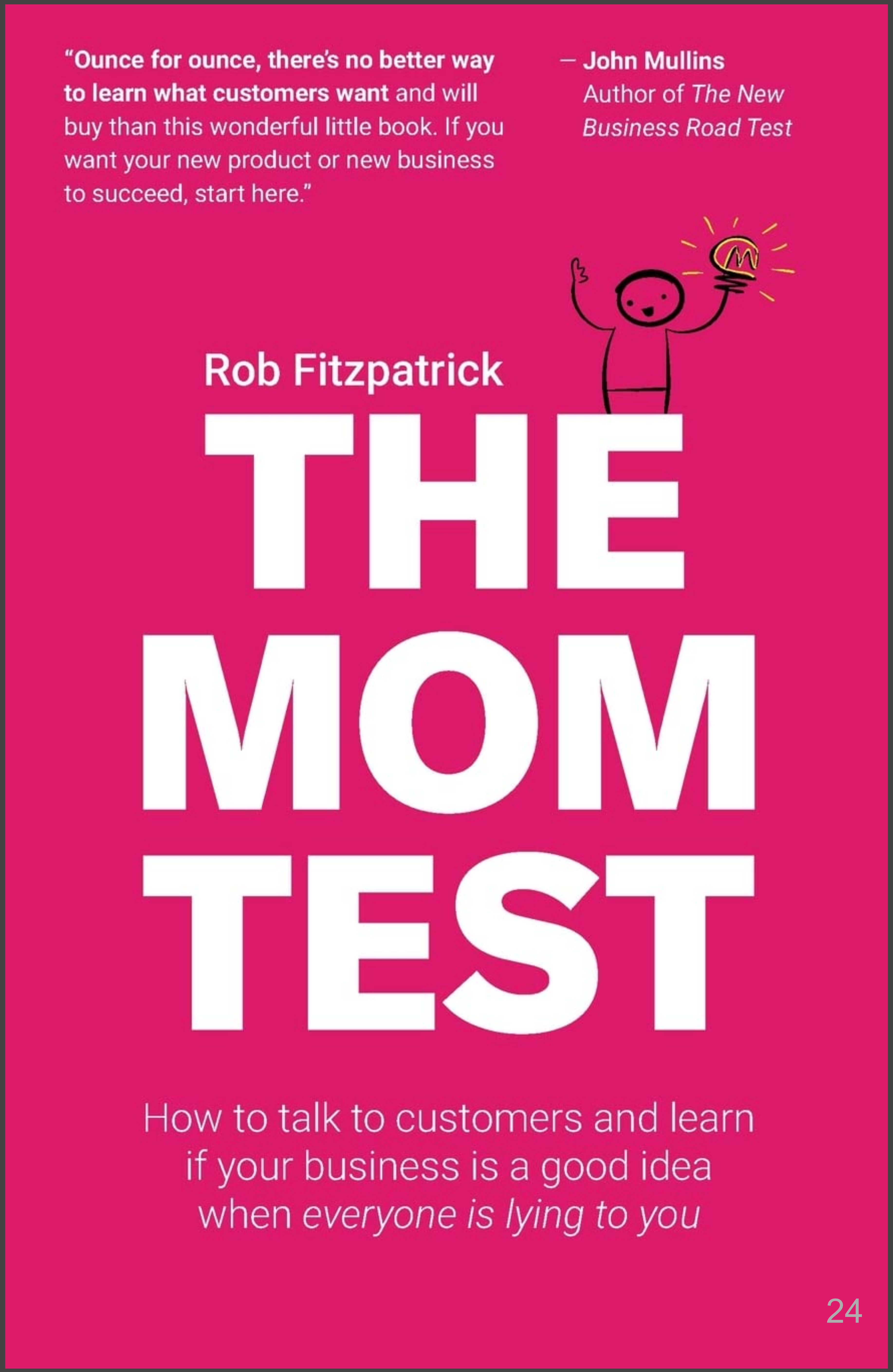
-

EN-speaking bias is real!

8 additional languages

# Reducing the error: Statement bias –

- Specific to C++
- What and How
- Which options



"**Ounce for ounce, there's no better way to learn what customers want** and will buy than this wonderful little book. If you want your new product or new business to succeed, start here."

— **John Mullins**
Author of *The New Business Road Test*

Rob Fitzpatrick

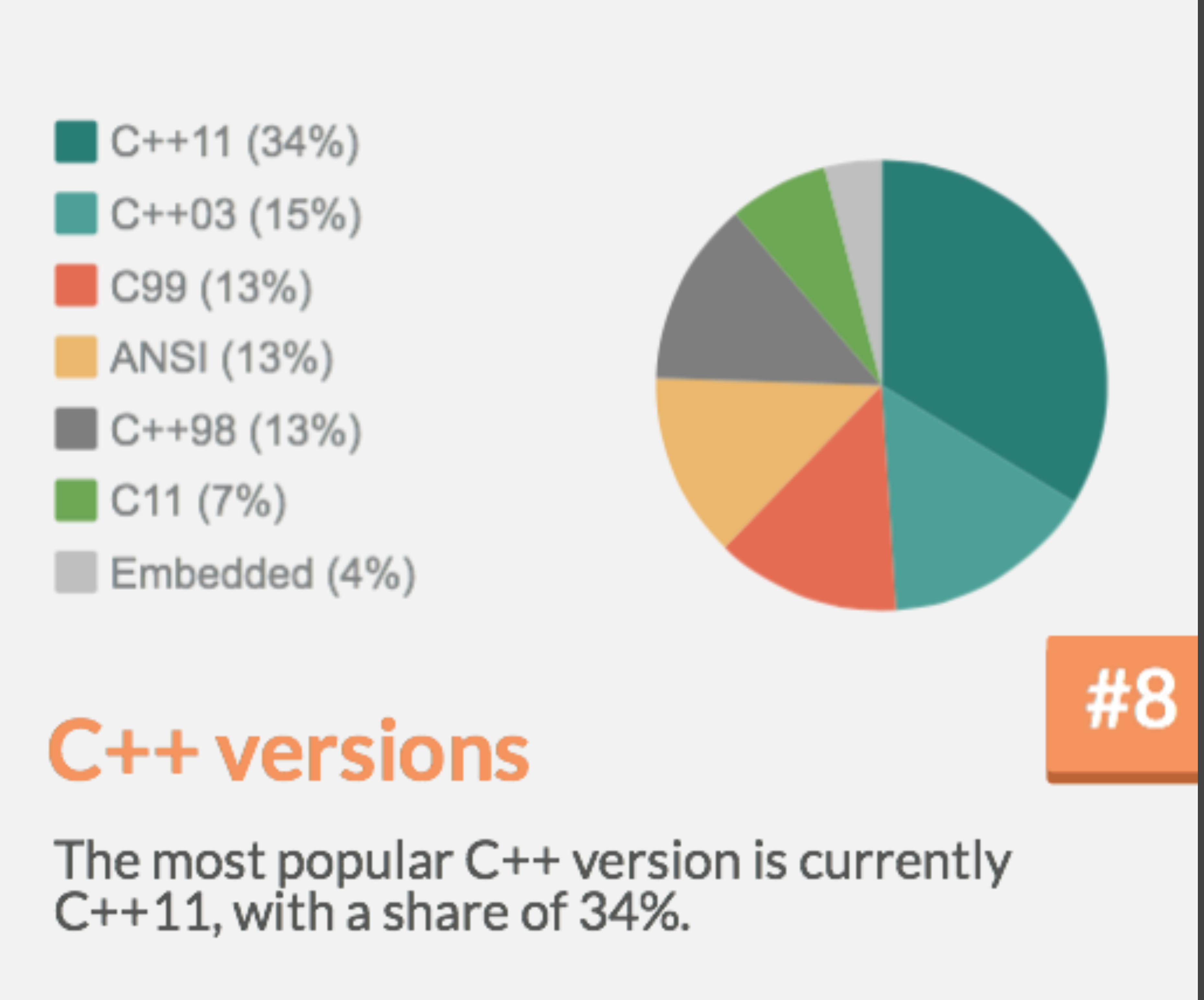
THE MOM TEST

How to talk to customers and learn if your business is a good idea when *everyone is lying to you*

# Asking questions is hard!

–

JetBRAINS 2015
C/C++ facts
we learned before bringing you
cross-platform IDE for C and C++
CLion

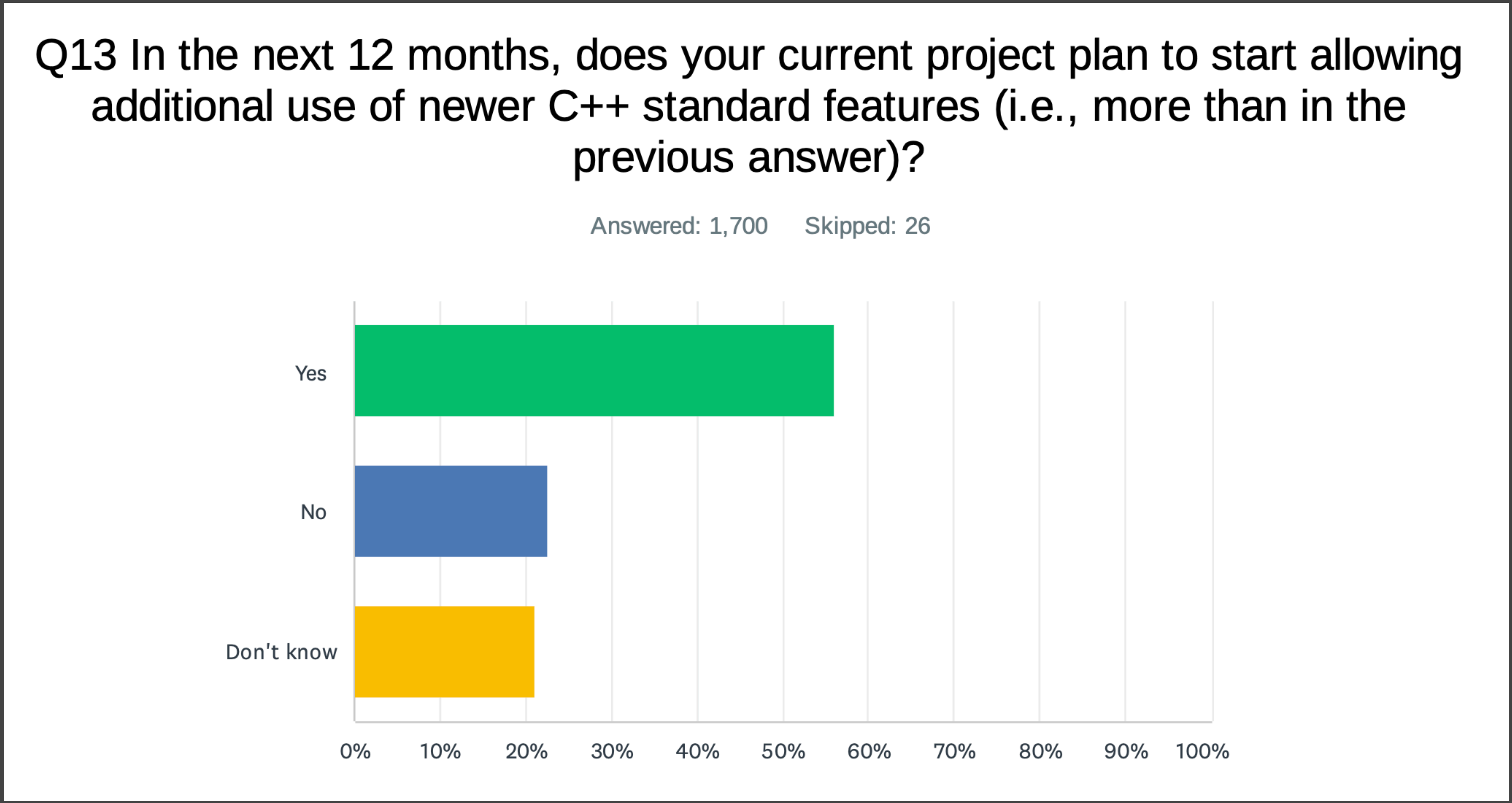


C++ versions

The most popular C++ version is currently C++11, with a share of 34%.

- 4.4mln C++ devs
- inconsistent list of industries
- C and C++ versions mixed

# Asking questions is hard!

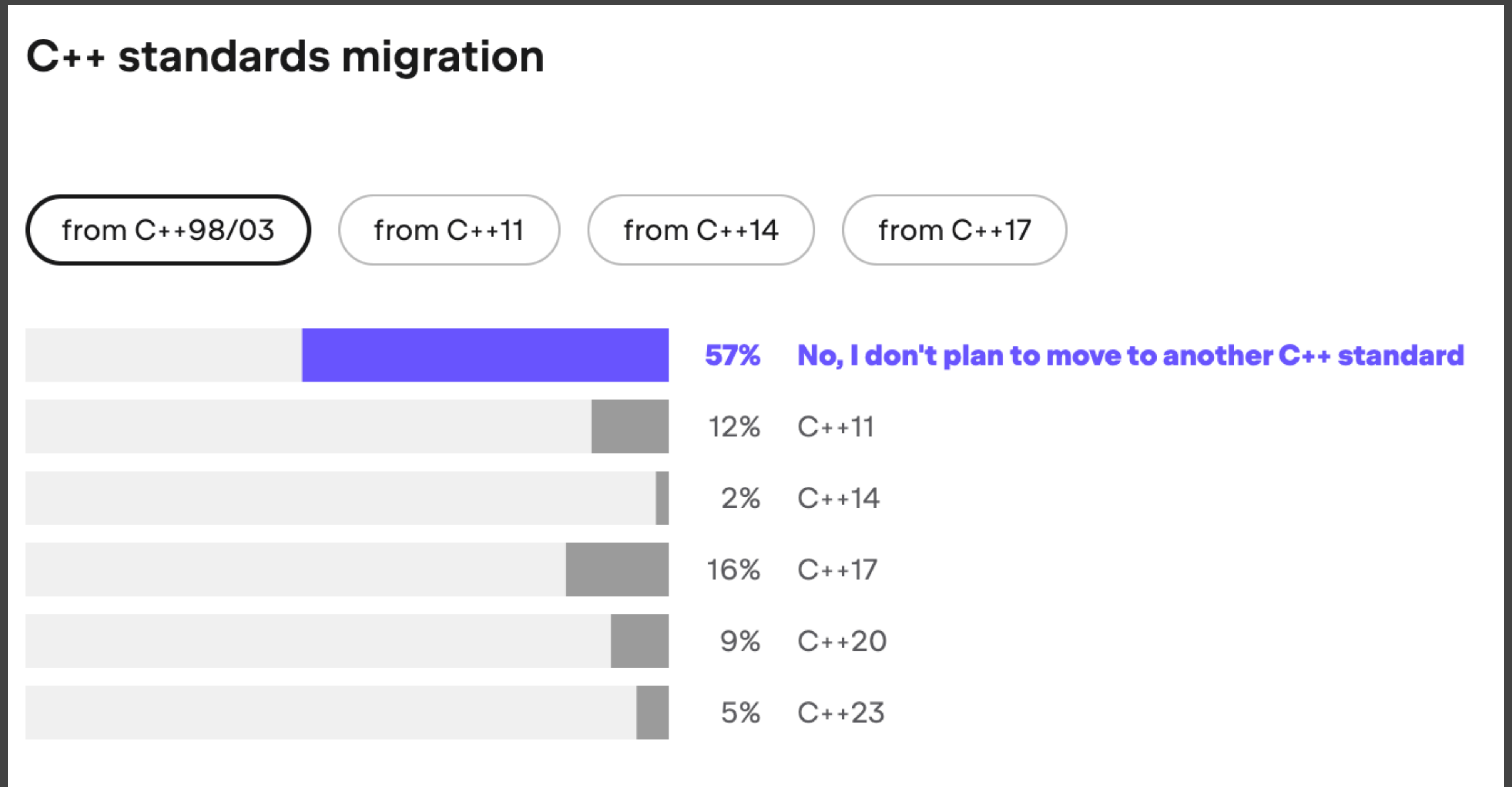


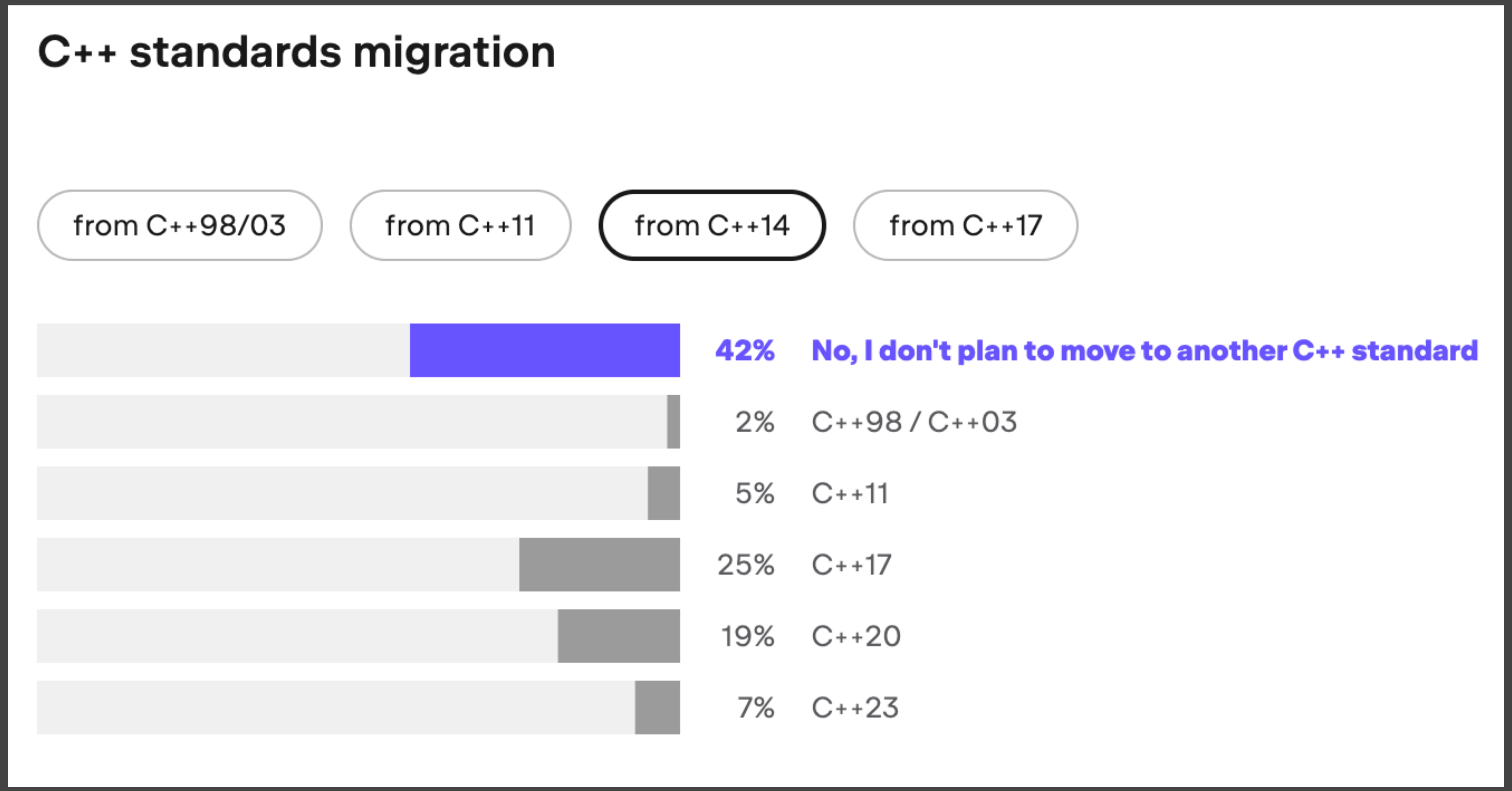
Q13 In the next 12 months, does your current project plan to start allowing additional use of newer C++ standard features (i.e., more than in the previous answer)?

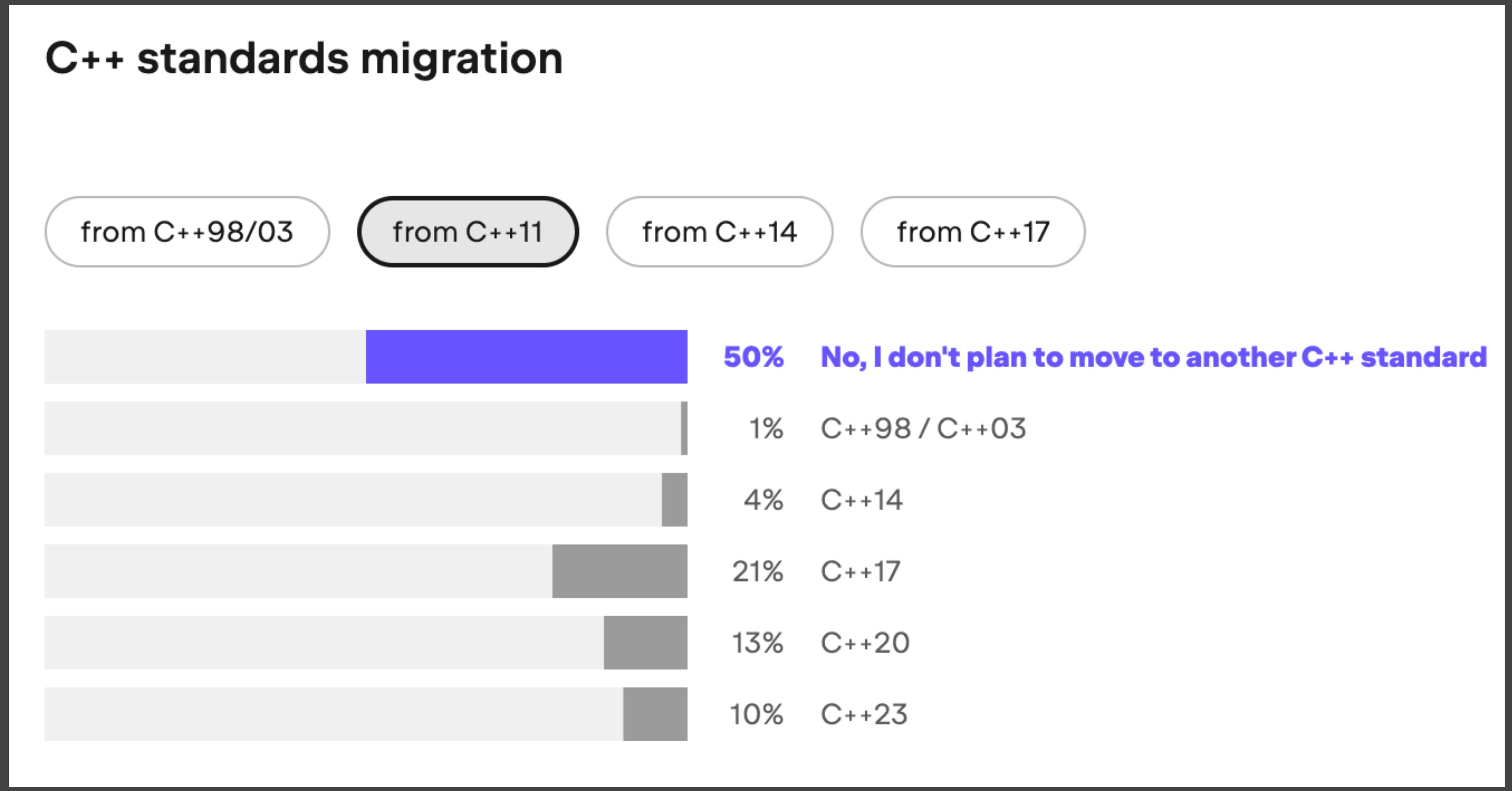
Answered: 1,700 Skipped: 26

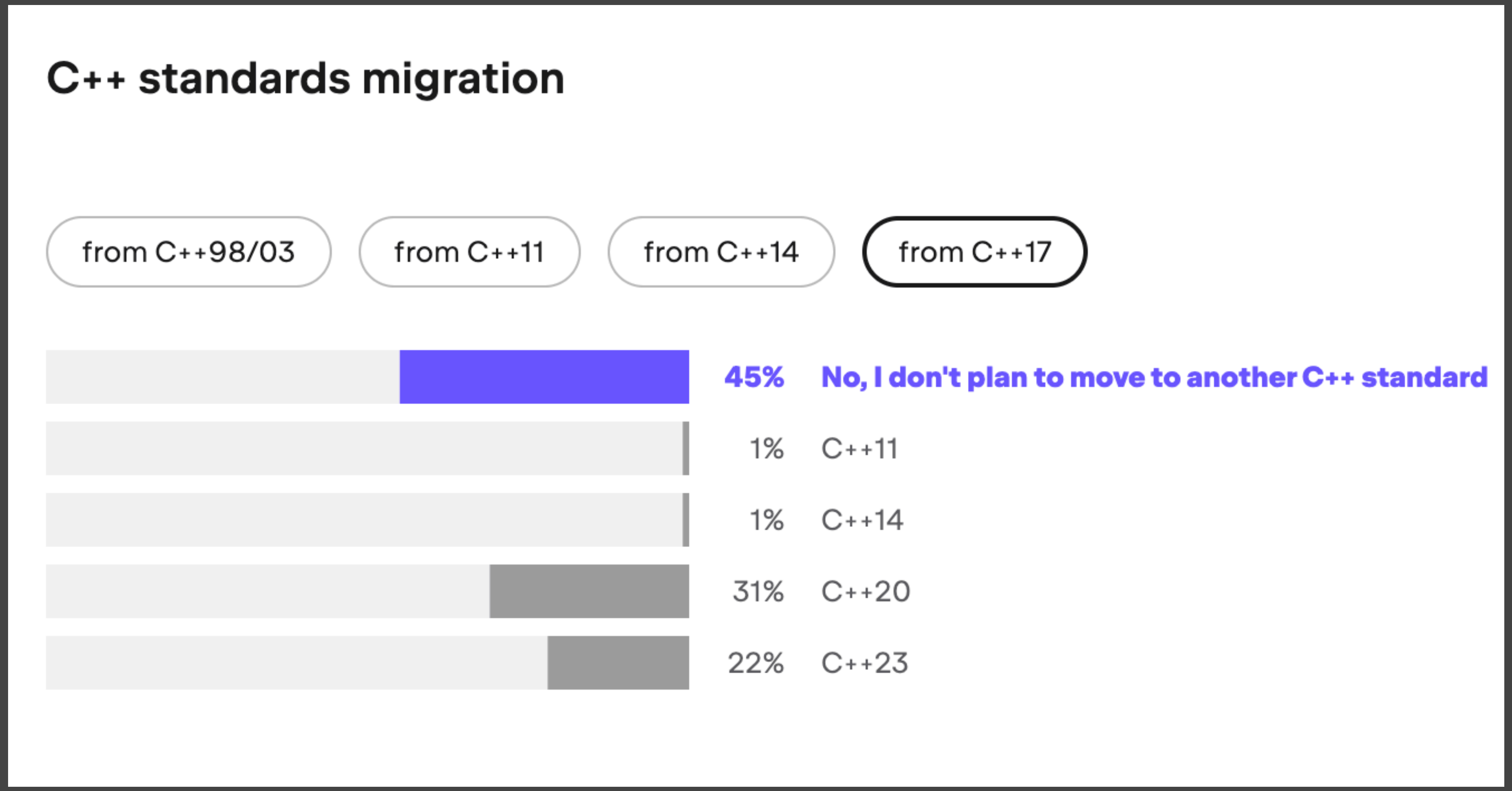
# Asking questions is hard!
_

# Upgrade paths are important

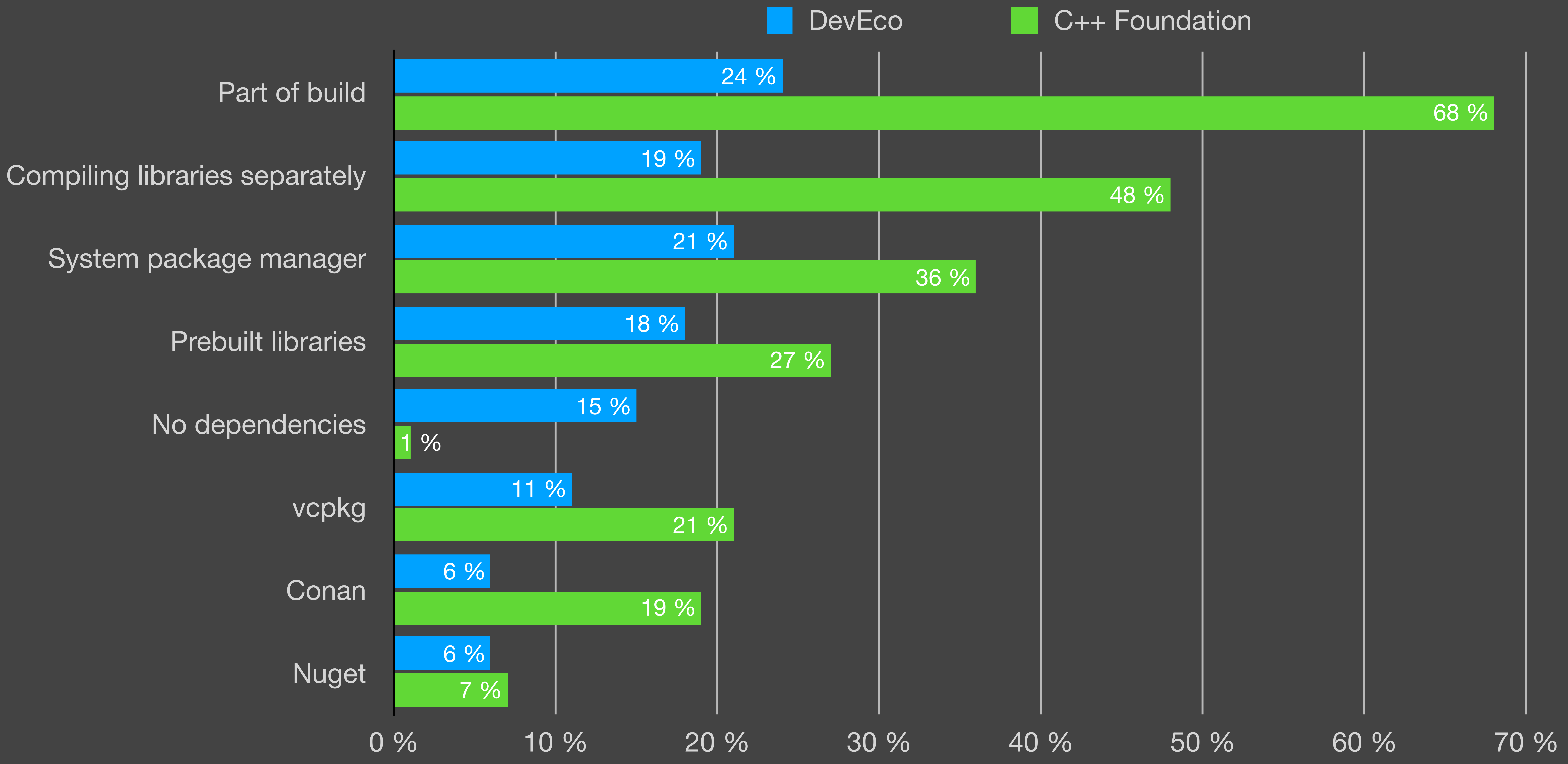






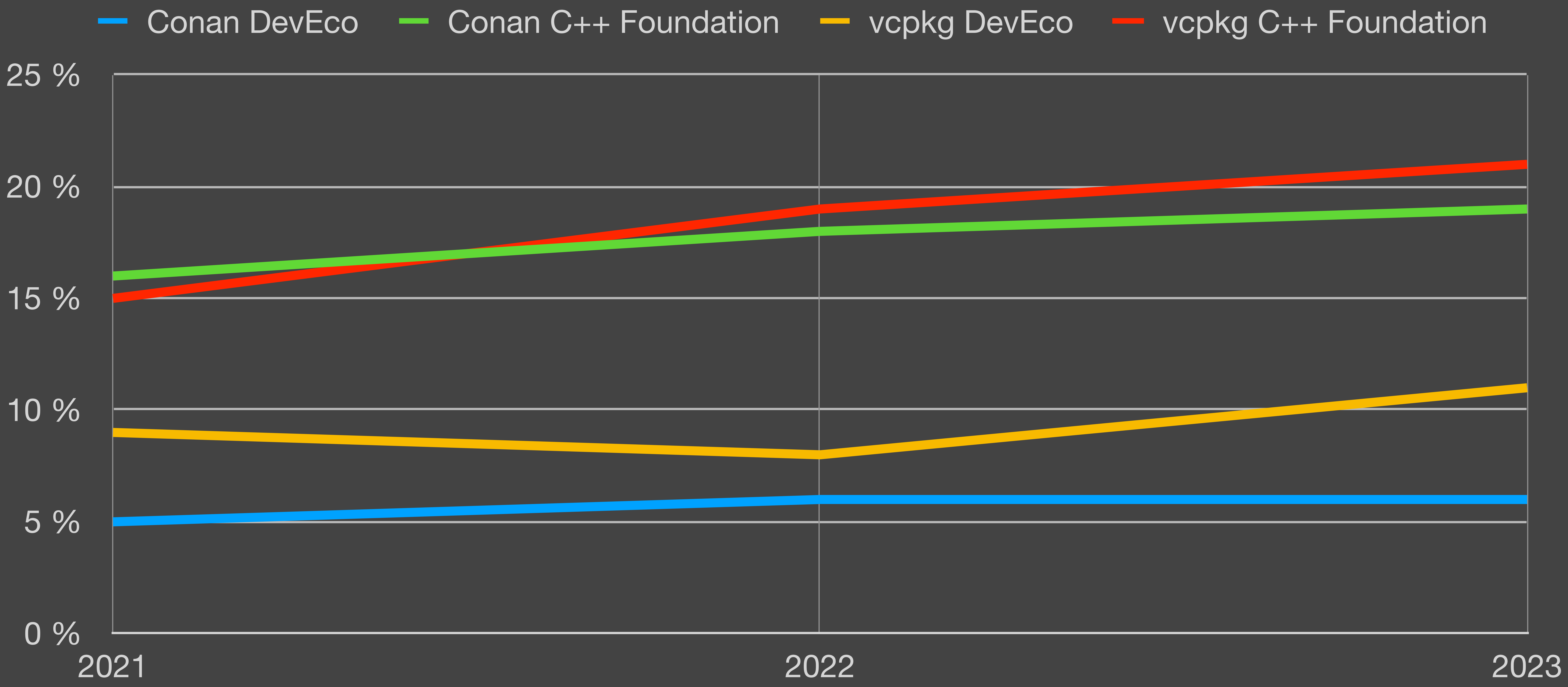


# Asking questions is hard!



**Check all that apply effect**

# Asking questions is hard!



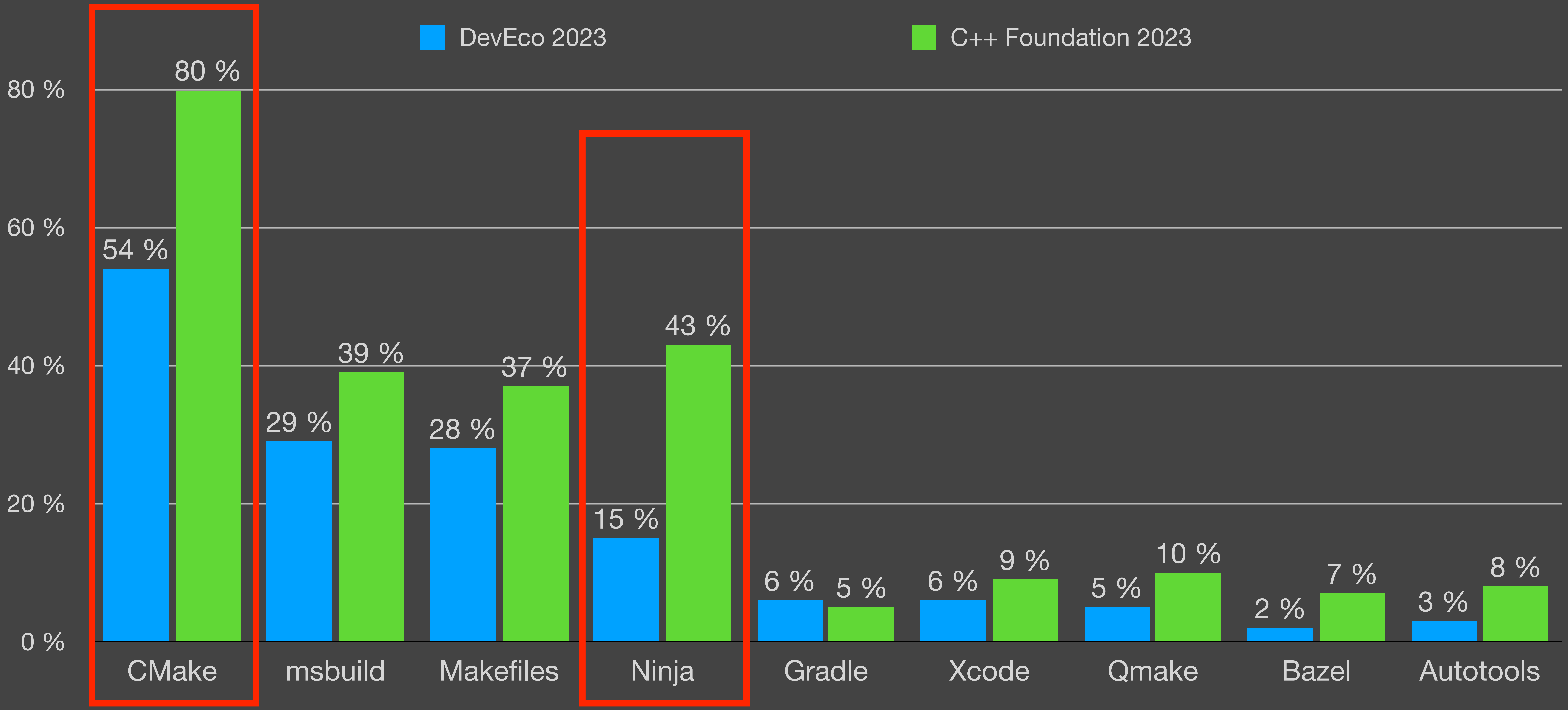
# Asking questions is hard!

-

“We have very solid internal indicators
(including telemetry, not only vanity metrics)
that prove that Conan keeps growing
solidly like more 20% YoY, and this year we have also seen
a lot of new very large enterprises onboarding Conan too,
so not only free users,
but also many new paying JFrog customers
grow every year. ”

Diego Rodriguez-Losada Gonzalez

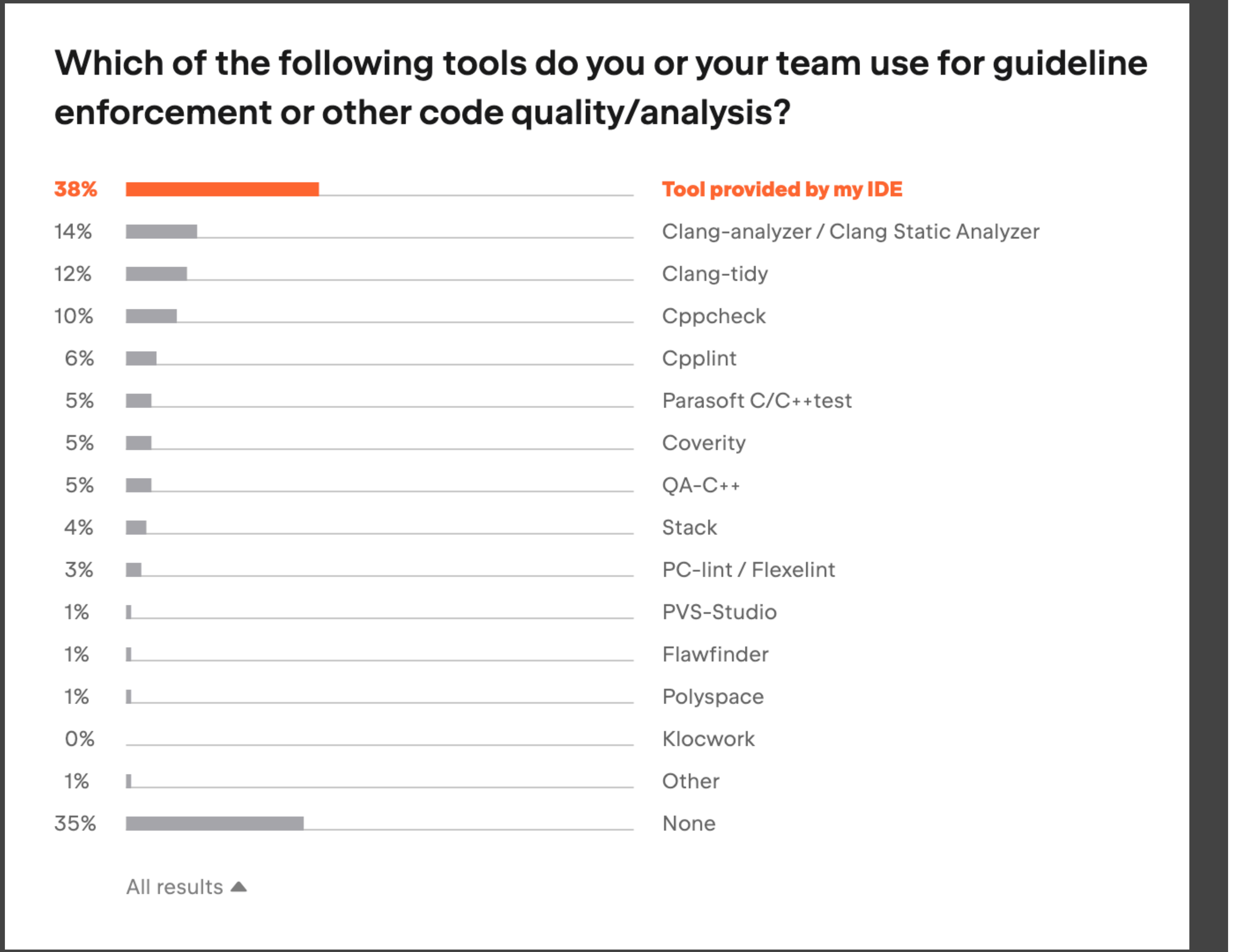
# Asking questions is hard!



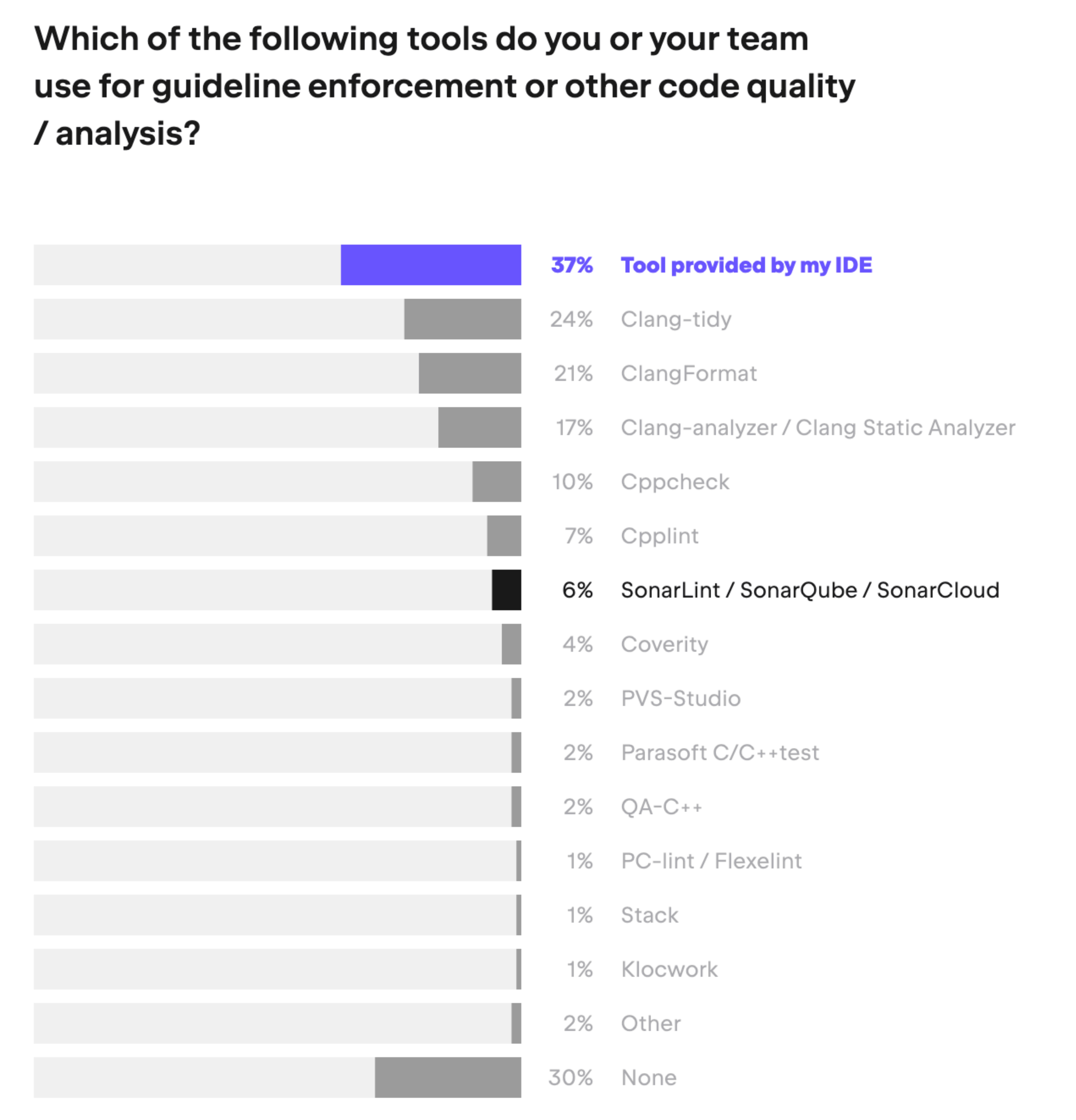
Check all that apply effect

Build tool vs Project model

# Asking questions is hard!



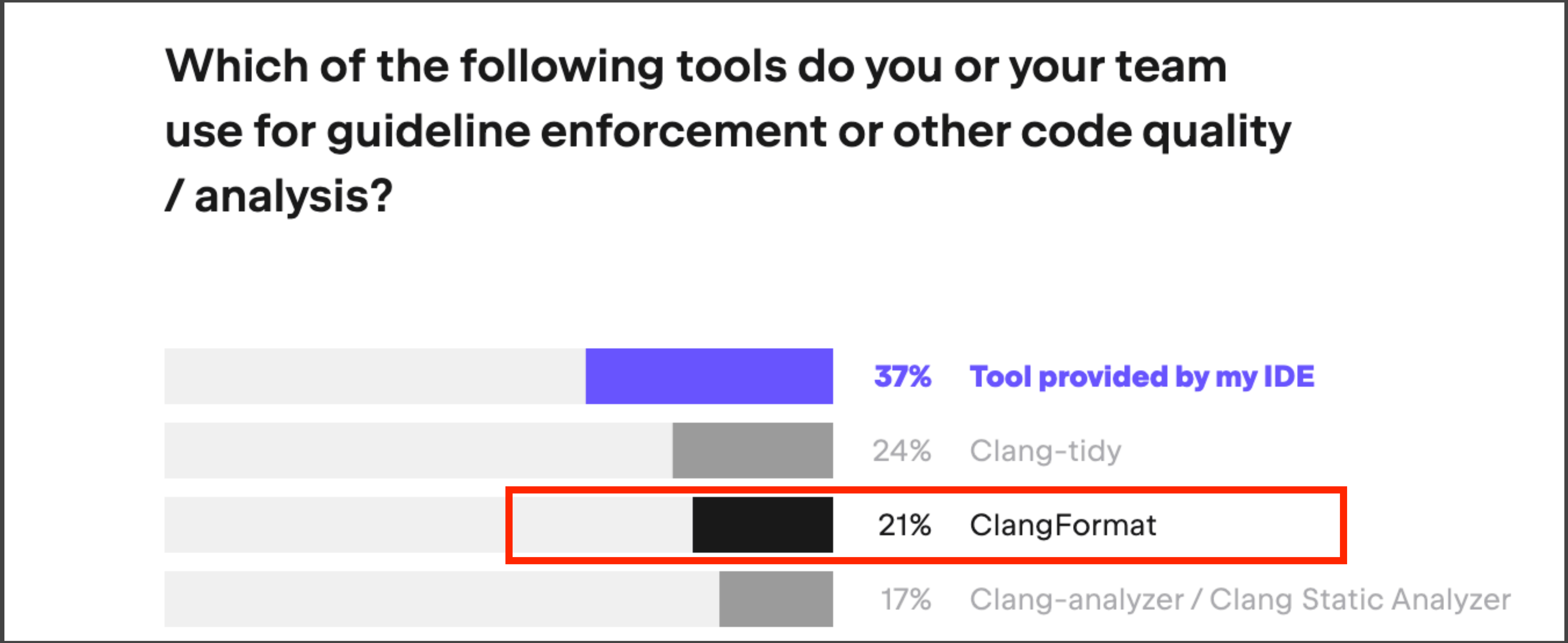
Which of the following tools do you or your team use for guideline enforcement or other code quality/analysis?



Which of the following tools do you or your team use for guideline enforcement or other code quality / analysis?

CI tools matter!

# Asking questions is hard!



Which of the following tools do you or your team use for guideline enforcement or other code quality / analysis?

## ClangFormat is not …
-

1. …a code analysis tool. But we treat it so.
2. …a standardize tooling. But we use it so.
3. …a true clang-based parser. But one sometimes thinks so.

## ClangFormat is not …

–

Scan GitHub for:

primary language C or C++ &

10+ stars &

latest version is scanned

⇒ 163,420 such repositories

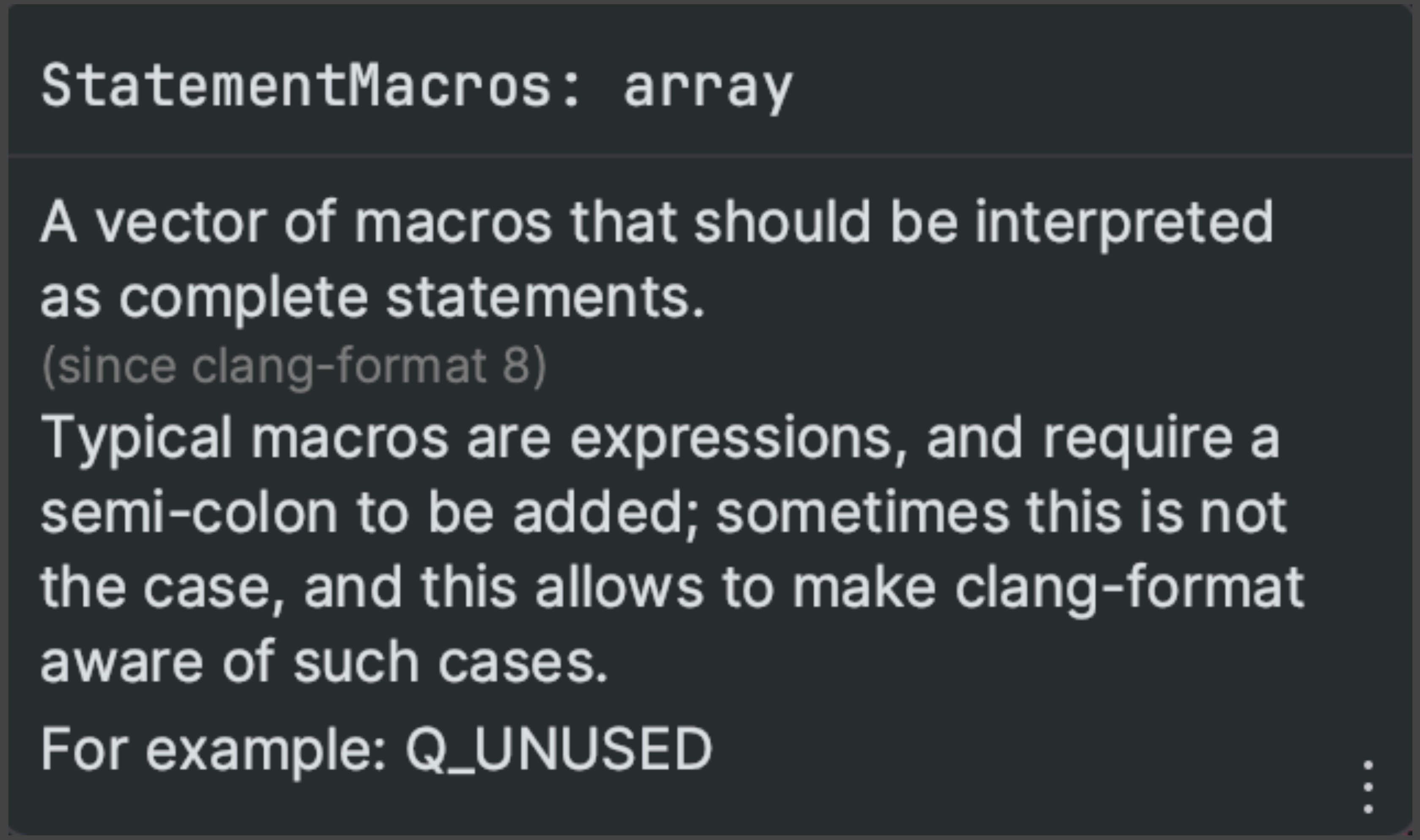
14,131 (8.65%) with .clang-format at any level

# ClangFormat and its fuzzy parser –

ClangFormat doesn’t parse…

…it generates annotated tokens

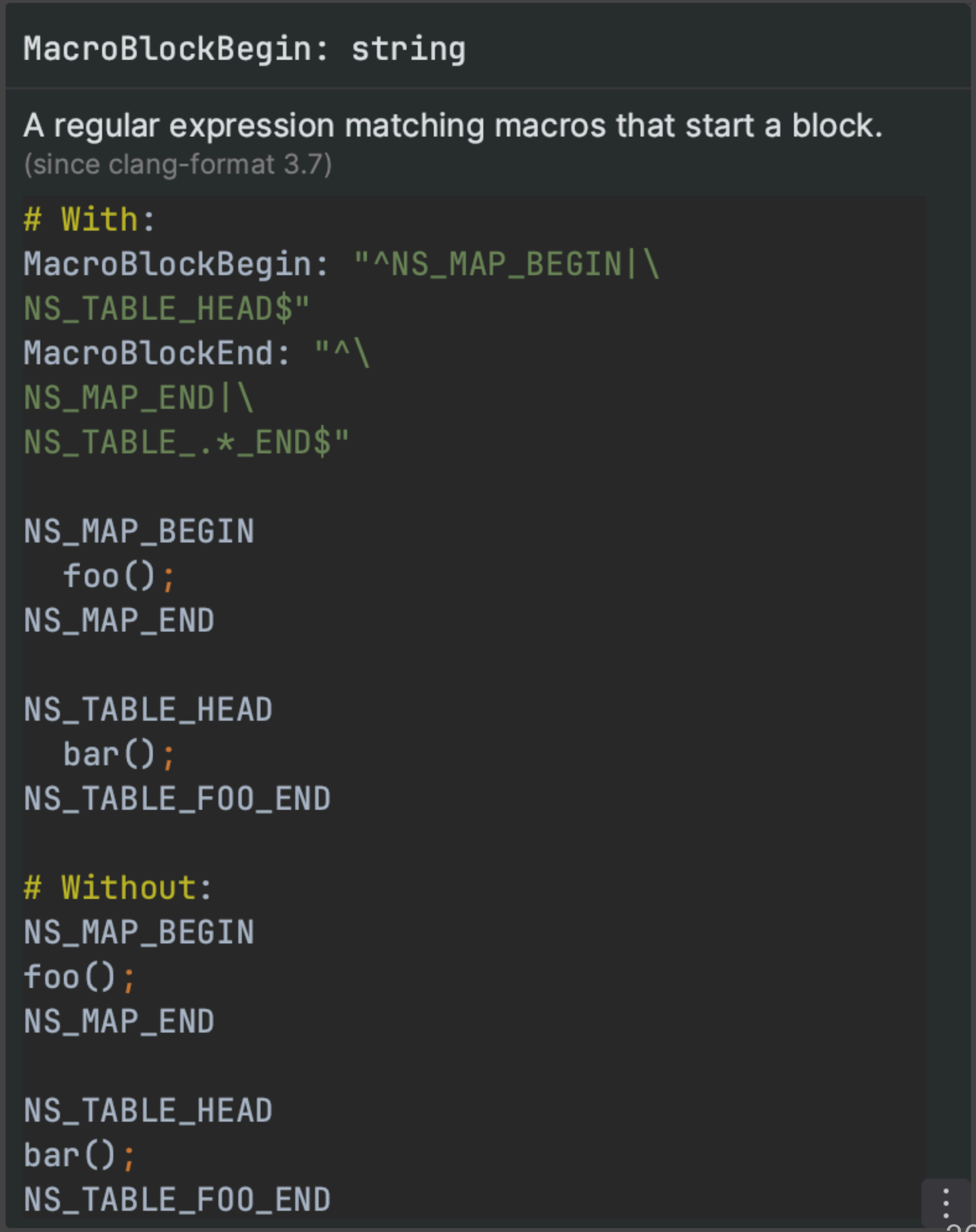
based on simple lexer.



`StatementMacros: array`

A vector of macros that should be interpreted as complete statements.
(since clang-format 8)
Typical macros are expressions, and require a semi-colon to be added; sometimes this is not the case, and this allows to make clang-format aware of such cases.

For example: Q_UNUSED

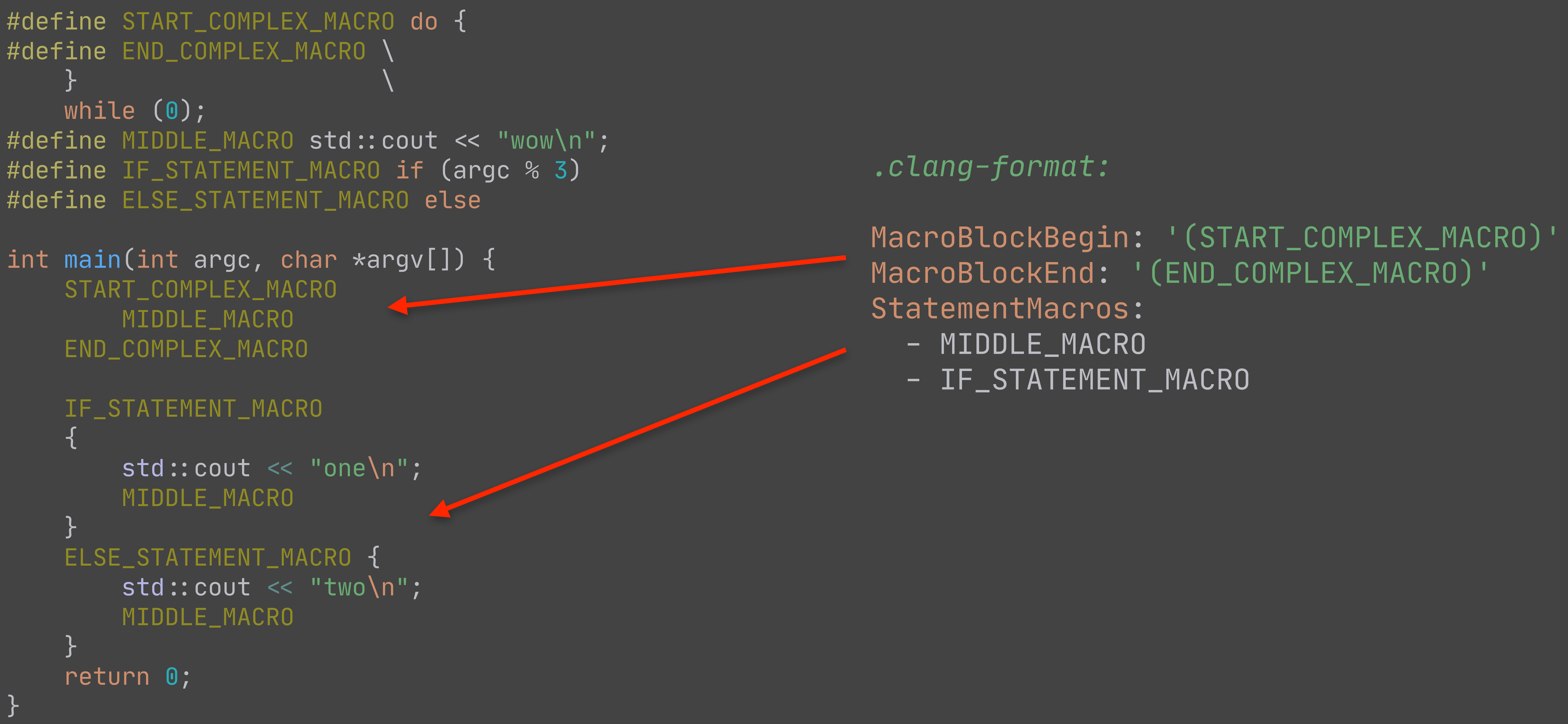


`MacroBlockBegin: string`

A regular expression matching macros that start a block.
(since clang-format 3.7)

```
# With:
MacroBlockBegin: "^NS_MAP_BEGIN|\
NS_TABLE_HEAD$"
MacroBlockEnd: "^\
NS_MAP_END|\
NS_TABLE_.*_END$"

NS_MAP_BEGIN
  foo();
NS_MAP_END

NS_TABLE_HEAD
  bar();
NS_TABLE_FOO_END

# Without:
NS_MAP_BEGIN
foo();
NS_MAP_END

NS_TABLE_HEAD
bar();
NS_TABLE_FOO_END
```

# ClangFormat and its fuzzy parser

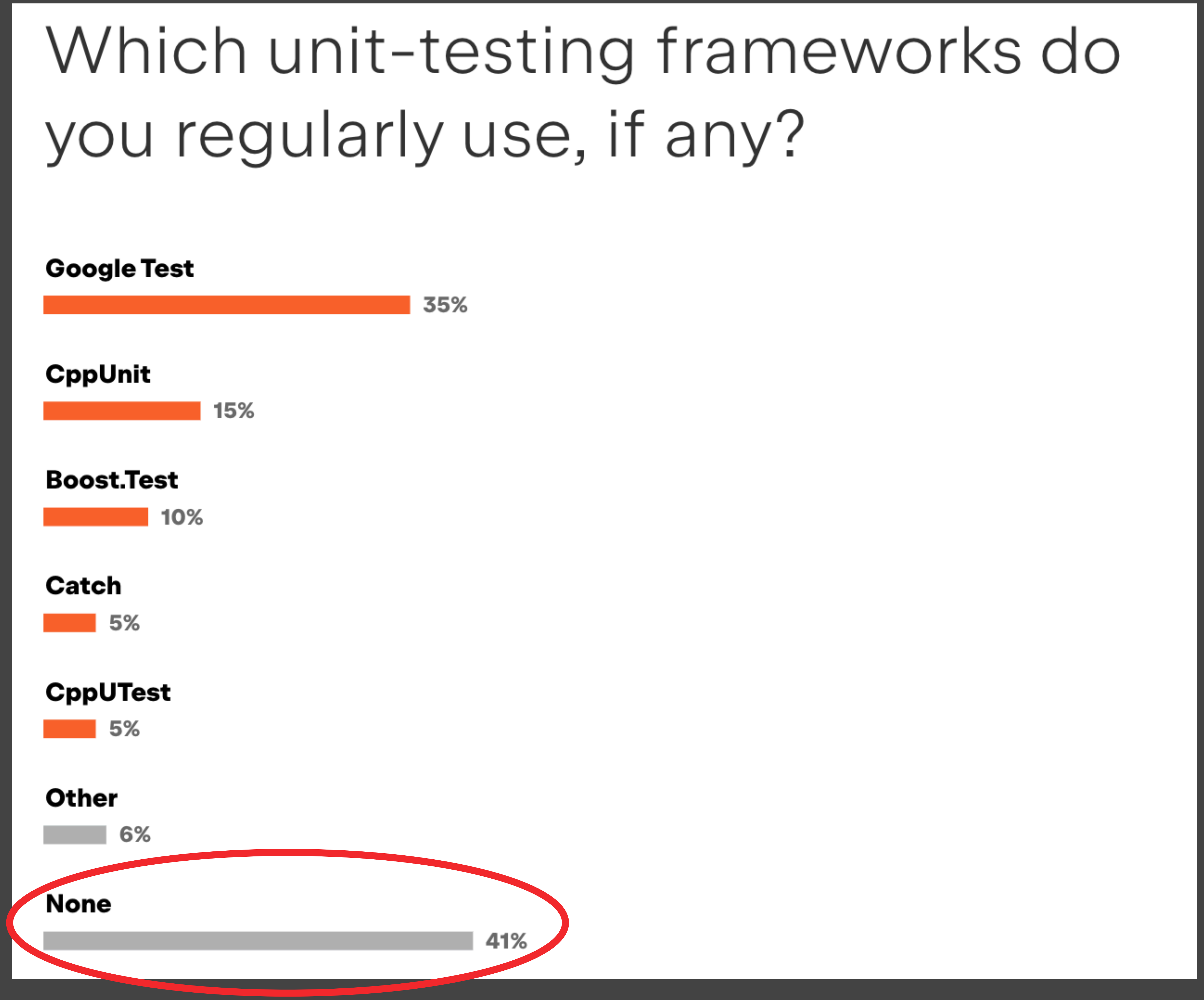


```cpp
#define START_COMPLEX_MACRO do {
#define END_COMPLEX_MACRO \
    }                     \
    while (0);
#define MIDDLE_MACRO std::cout << "wow\n";
#define IF_STATEMENT_MACRO if (argc % 3)
#define ELSE_STATEMENT_MACRO else

int main(int argc, char *argv[]) {
    START_COMPLEX_MACRO
        MIDDLE_MACRO
    END_COMPLEX_MACRO

    IF_STATEMENT_MACRO
    {
        std::cout << "one\n";
        MIDDLE_MACRO
    }
    ELSE_STATEMENT_MACRO {
        std::cout << "two\n";
        MIDDLE_MACRO
    }
    return 0;
}
```

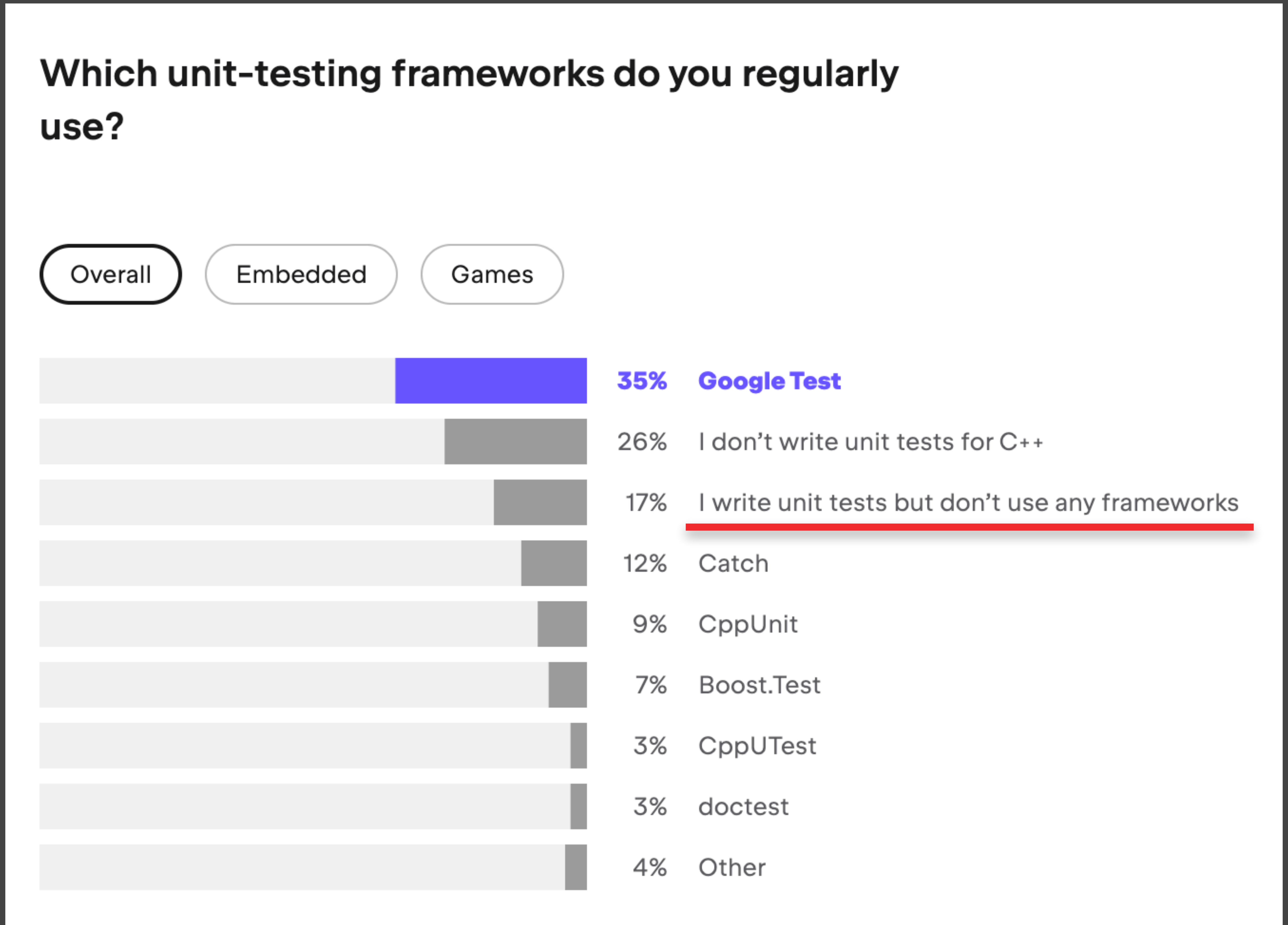
*.clang-format:*

```yaml
MacroBlockBegin: '(START_COMPLEX_MACRO)'
MacroBlockEnd: '(END_COMPLEX_MACRO)'
StatementMacros:
  - MIDDLE_MACRO
  - IF_STATEMENT_MACRO
```

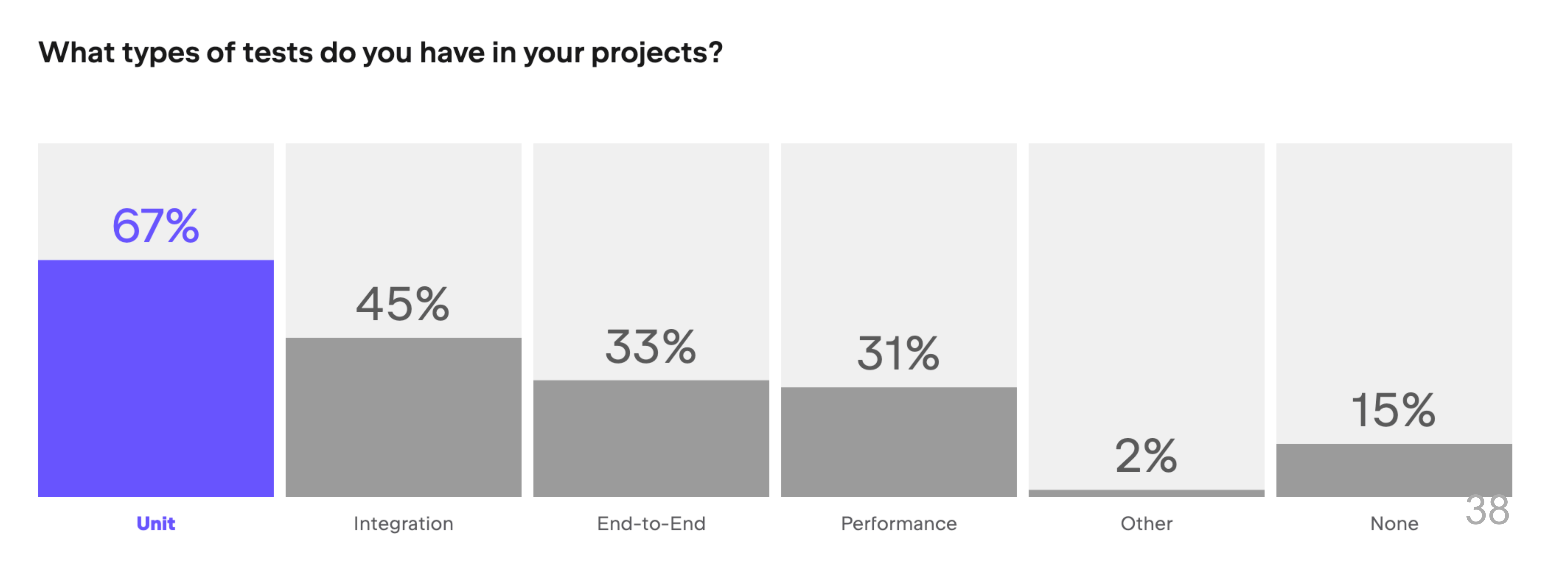
# Asking questions is hard!



Which unit-testing frameworks do you regularly use, if any?



Which unit-testing frameworks do you regularly use?



What types of tests do you have in your projects?

No Unit Tests ≠ No Tests

# Asking questions is hard! Summary

1. C is not C++
2. C++ versions paths
3. Check all that apply effect
4. Package management
5. Project model vs Build tool
6. Code Analysis on CI matters
7. ClangFormat is not <…> but we use it so
8. No Unit Tests No Tests

# Validation
-

How to validate data?

# Validation = probative comparison

1. Compare on the same audience slices
2. Compare on the same questions (1-1 msg match)
3. Explain the difference

# Validation = probative comparison

| Dev Eco 2023 | C++ Foundation 2023 |
|---|---|
| How do you manage your third party libraries in C++? | How do you manage your C++ 1st and 3rd party libraries? |
| Which project models or build systems do you regularly use, if any? | What build tools do you use? |
| Which C++ standard(s) do you regularly use? | What version(s) of C++ are you allowed to use on your current project (work or school)? |
| Do you plan to move from <…> to another C++ standard in the next 12 months? | In the next 12 months, does your current project plan to start allowing additional use of newer C++ standard features |
| Is your current project planning to use any of the following C++20 features in the next 12 months? | In the next 12 months, does your current project plan to allow use of these C++20 features in production code? |

## Thank you to

—

1. JetBrains Market Research & Analytics Team for all their work
2. Herb Sutter for collaboration around our two surveys
3. Diego Rodriguez-Losada Gonzalez, Inbal Levi, and Bryce Adelstein Lelbach for reviewing and commenting on 2023 data
4. Inbal Levi for inspiring me to do this talk

# Surveys

1. JetBrains DevEco since 2017, https://www.jetbrains.com/lp/devecosystem-2022/
2. Annual C++ Developer Survey "Lite", https://isocpp.org/files/papers/CppDevSurvey-2023-summary.pdf
3. Meeting C++ Survey, https://meetingcpp.com/mcpp/survey/
4. The Mom Test, https://www.amazon.com/Mom-Test-customers-business-everyone/dp/1492180742/ref=tmm_pap_swatch_0
5. C++ facts we learned before doing CLion, https://blog.jetbrains.com/clion/2015/07/infographics-cpp-facts-before-clion/