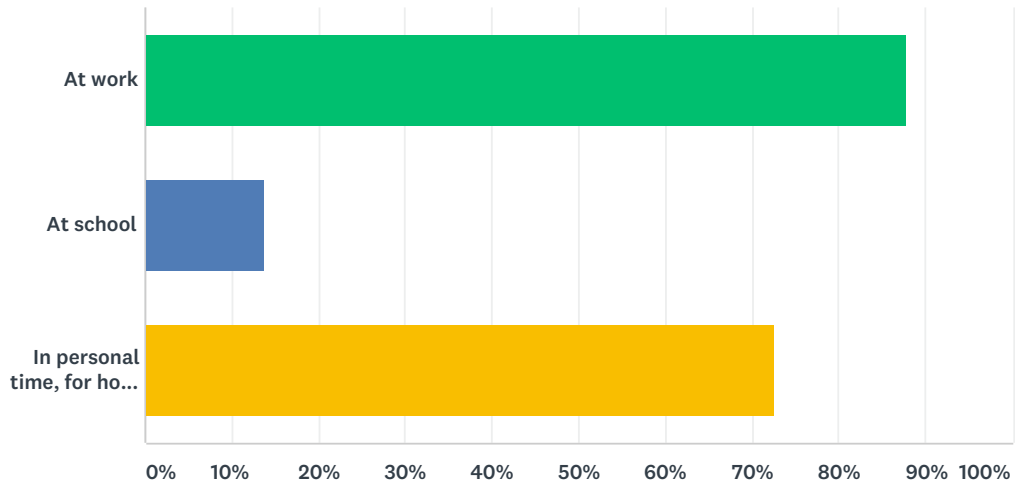


## Q1 Where do you use C++? (select all that apply)

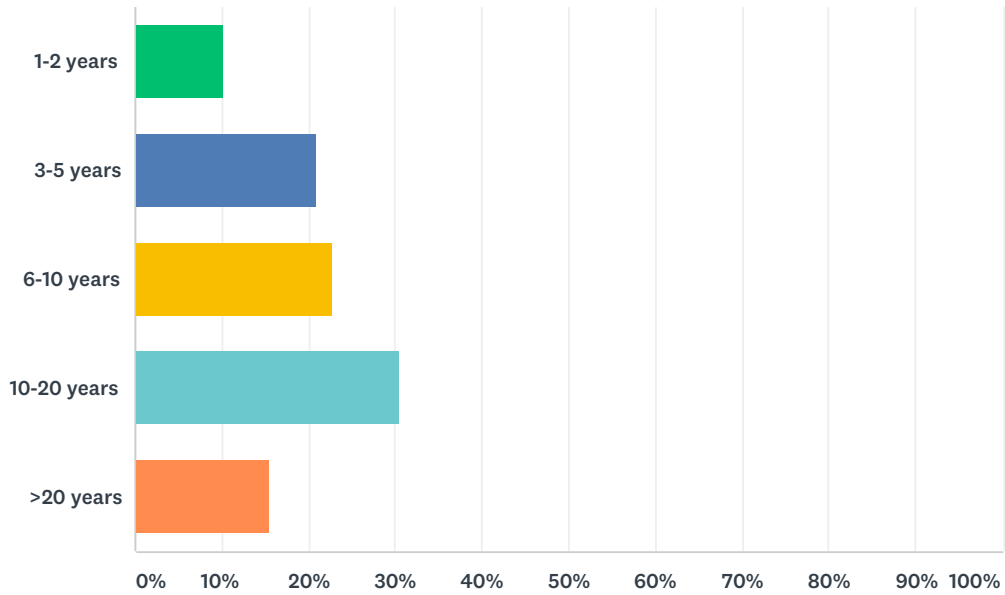
Answered: 3,280 Skipped: 6



ANSWER CHOICES	RESPONSES	
At work	87.93%	2,884
At school	13.81%	453
In personal time, for hobby projects or to try new things	72.56%	2,380
Total Respondents: 3,280		

## Q2 How many years of programming experience do you have in C++ specifically?

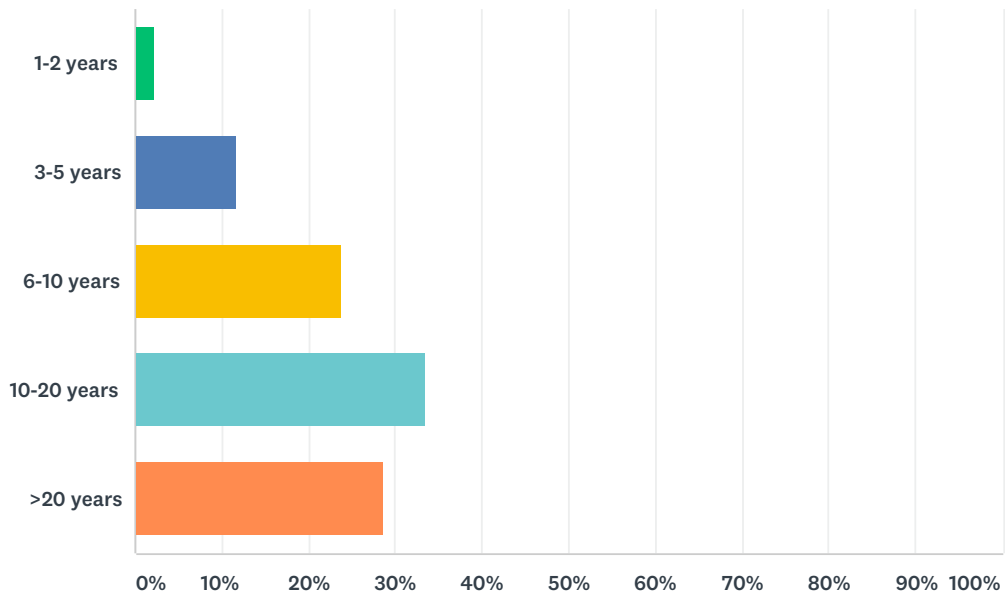
Answered: 3,271 Skipped: 15



ANSWER CHOICES	RESPONSES	
1-2 years	10.27%	336
3-5 years	20.85%	682
6-10 years	22.87%	748
10-20 years	30.63%	1,002
>20 years	15.38%	503
TOTAL		3,271

### Q3 How many years of programming experience do you have overall (all languages)?

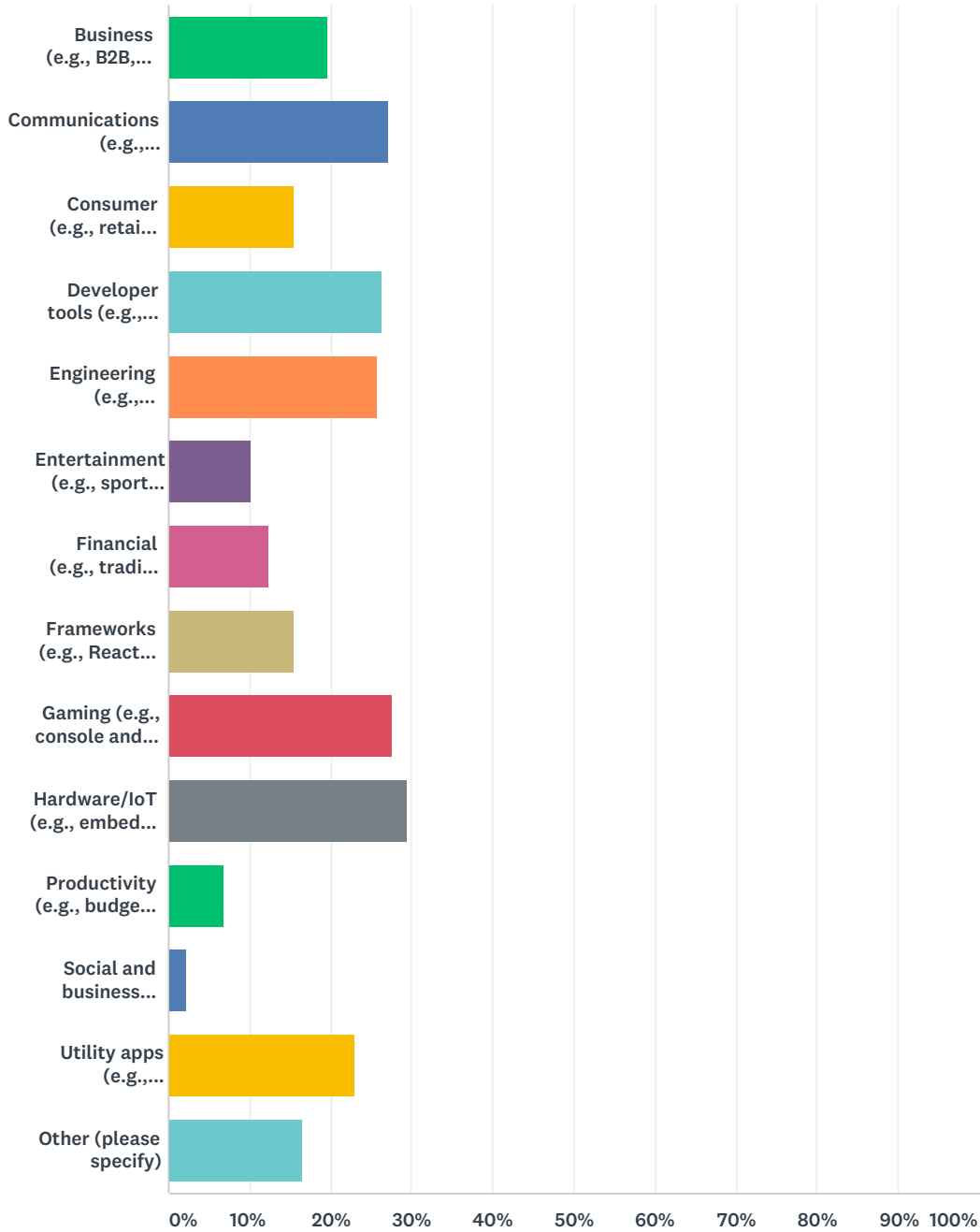
Answered: 3,276 Skipped: 10



ANSWER CHOICES	RESPONSES
1-2 years	2.26% 74
3-5 years	11.66% 382
6-10 years	23.78% 779
10-20 years	33.55% 1,099
>20 years	28.75% 942
TOTAL	3,276

## Q4 What types of projects do you work on? (select all that apply)

Answered: 3,269 Skipped: 17



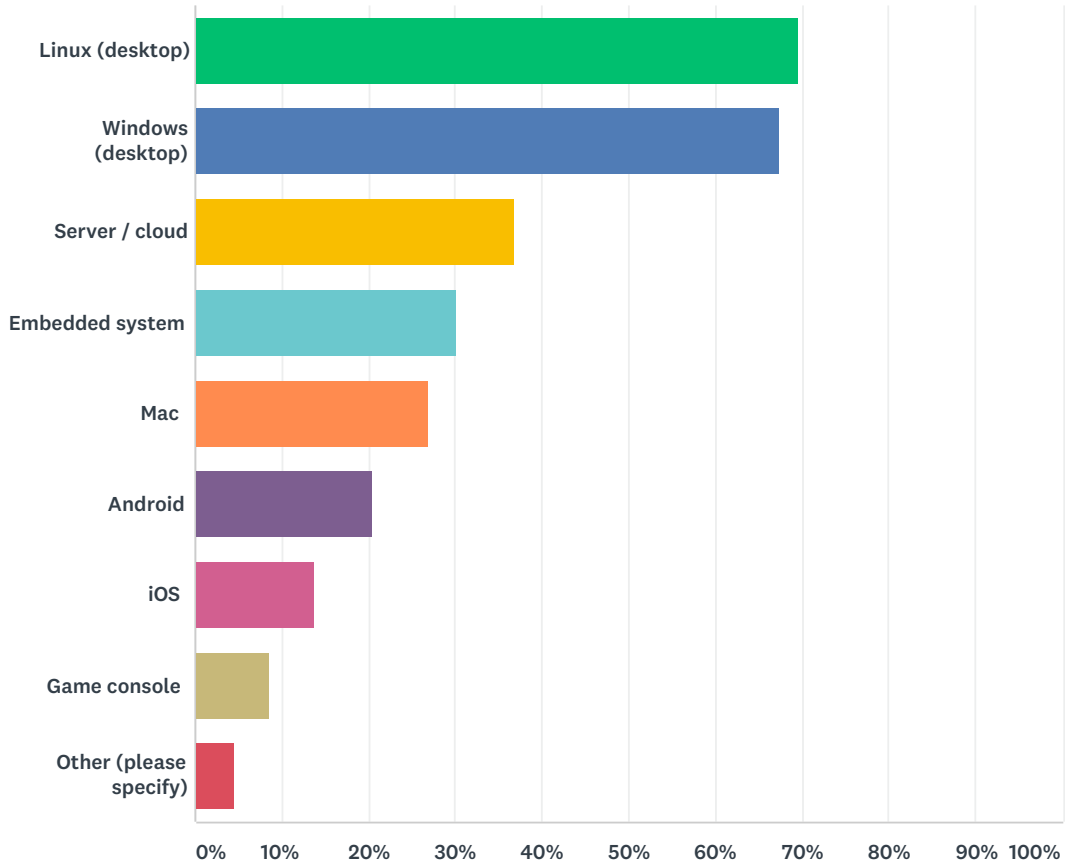
ANSWER CHOICES	RESPONSES	
Business (e.g., B2B, B2E)	19.64%	642
Communications (e.g., networking, email)	27.23%	890
Consumer (e.g., retail websites, mobile apps)	15.42%	504
Developer tools (e.g., compilers, code editors)	26.31%	860
Engineering (e.g., avionics, power management)	25.76%	842

## C++ Developer Survey "Lite": 2018-02

Entertainment (e.g., sports apps, video streaming)	10.25%	335
Financial (e.g., trading, mortgage, asset management)	12.30%	402
Frameworks (e.g., React, Unity)	15.57%	509
Gaming (e.g., console and mobile games)	27.62%	903
Hardware/IoT (e.g., embedded systems, home automation)	29.40%	961
Productivity (e.g., budget tracking, note taking)	6.91%	226
Social and business networking (e.g., Facebook, Twitter)	2.32%	76
Utility apps (e.g., calculators, simple image editors)	22.94%	750
Other (please specify)	16.49%	539
Total Respondents: 3,269		

## Q5 What platforms do you develop for? (select all that apply)

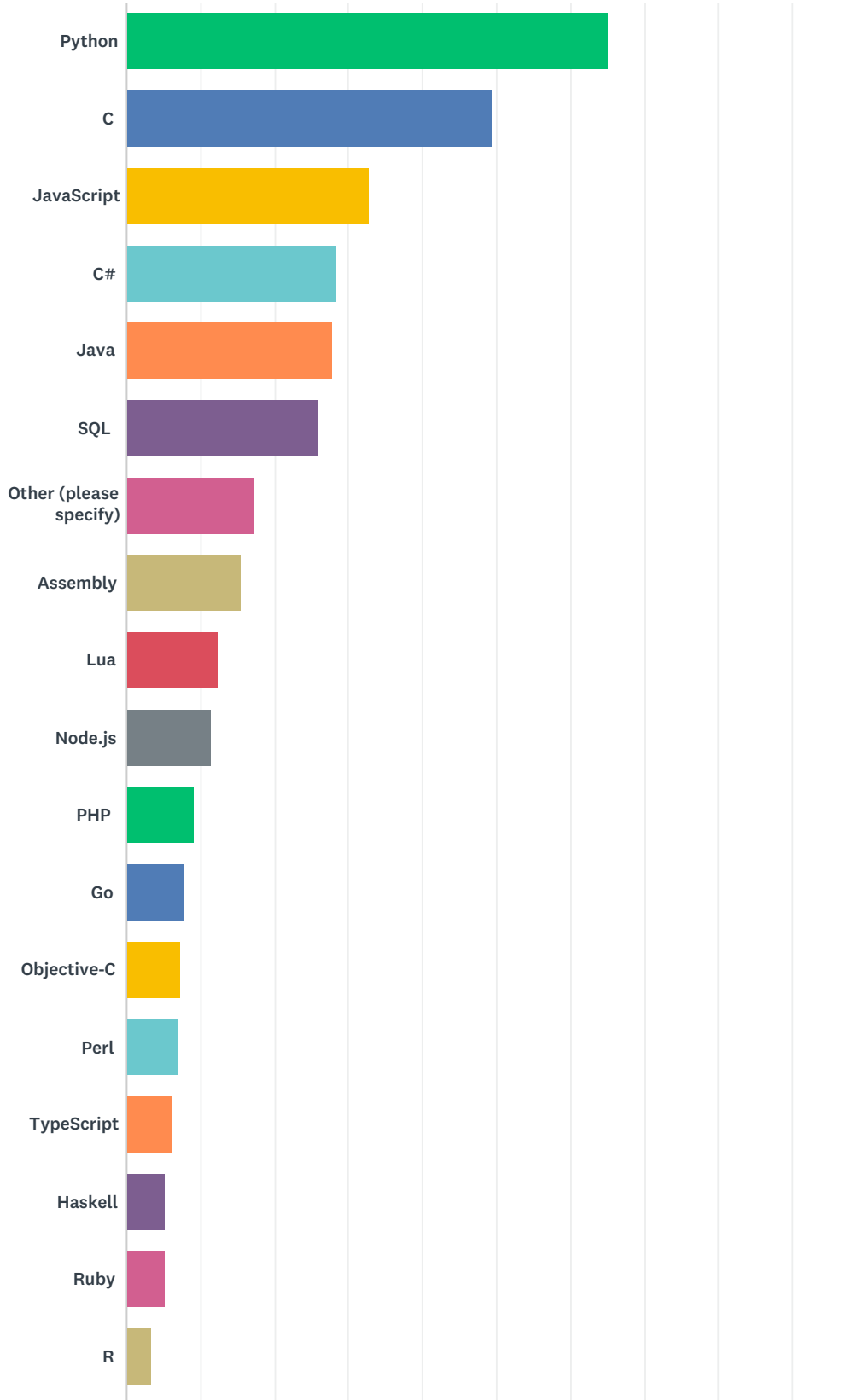
Answered: 3,271 Skipped: 15



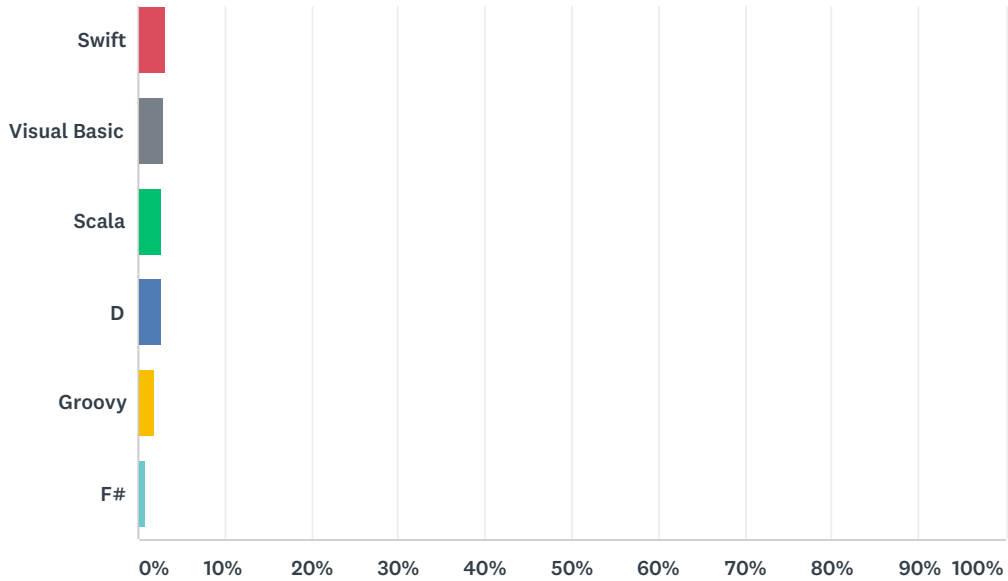
ANSWER CHOICES	RESPONSES	
Linux (desktop)	69.76%	2,282
Windows (desktop)	67.44%	2,206
Server / cloud	36.78%	1,203
Embedded system	30.08%	984
Mac	26.90%	880
Android	20.57%	673
iOS	13.76%	450
Game console	8.62%	282
Other (please specify)	4.65%	152
Total Respondents: 3,271		

# Q6 Besides C++, what programming languages/environments do you use in your current and recent projects? (select all that apply)

Answered: 3,190 Skipped: 96



## C++ Developer Survey "Lite": 2018-02



ANSWER CHOICES	RESPONSES	
Python	65.08%	2,076
C	49.40%	1,576
JavaScript	32.82%	1,047
C#	28.53%	910
Java	27.74%	885
SQL	25.89%	826
Other (please specify)	17.40%	555
Assembly	15.45%	493
Lua	12.41%	396
Node.js	11.41%	364
PHP	9.18%	293
Go	7.96%	254
Objective-C	7.27%	232
Perl	7.15%	228
TypeScript	6.27%	200
Haskell	5.30%	169
Ruby	5.17%	165
R	3.45%	110
Swift	3.20%	102
Visual Basic	2.92%	93
Scala	2.70%	86
D	2.66%	85

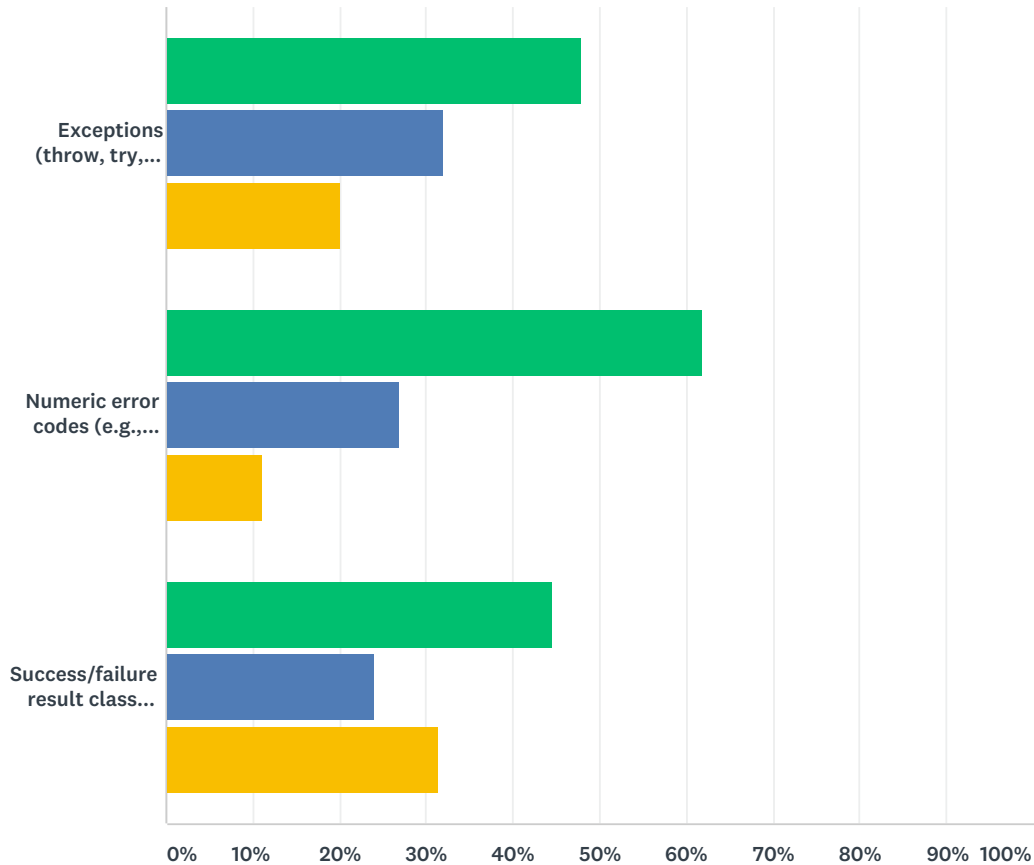


## C++ Developer Survey "Lite": 2018-02

Groovy	1.82%	58
F#	0.78%	25
Total Respondents: 3,190		

## Q7 What error reporting methods are allowed on your current project (work or school)?

Answered: 3,255 Skipped: 31

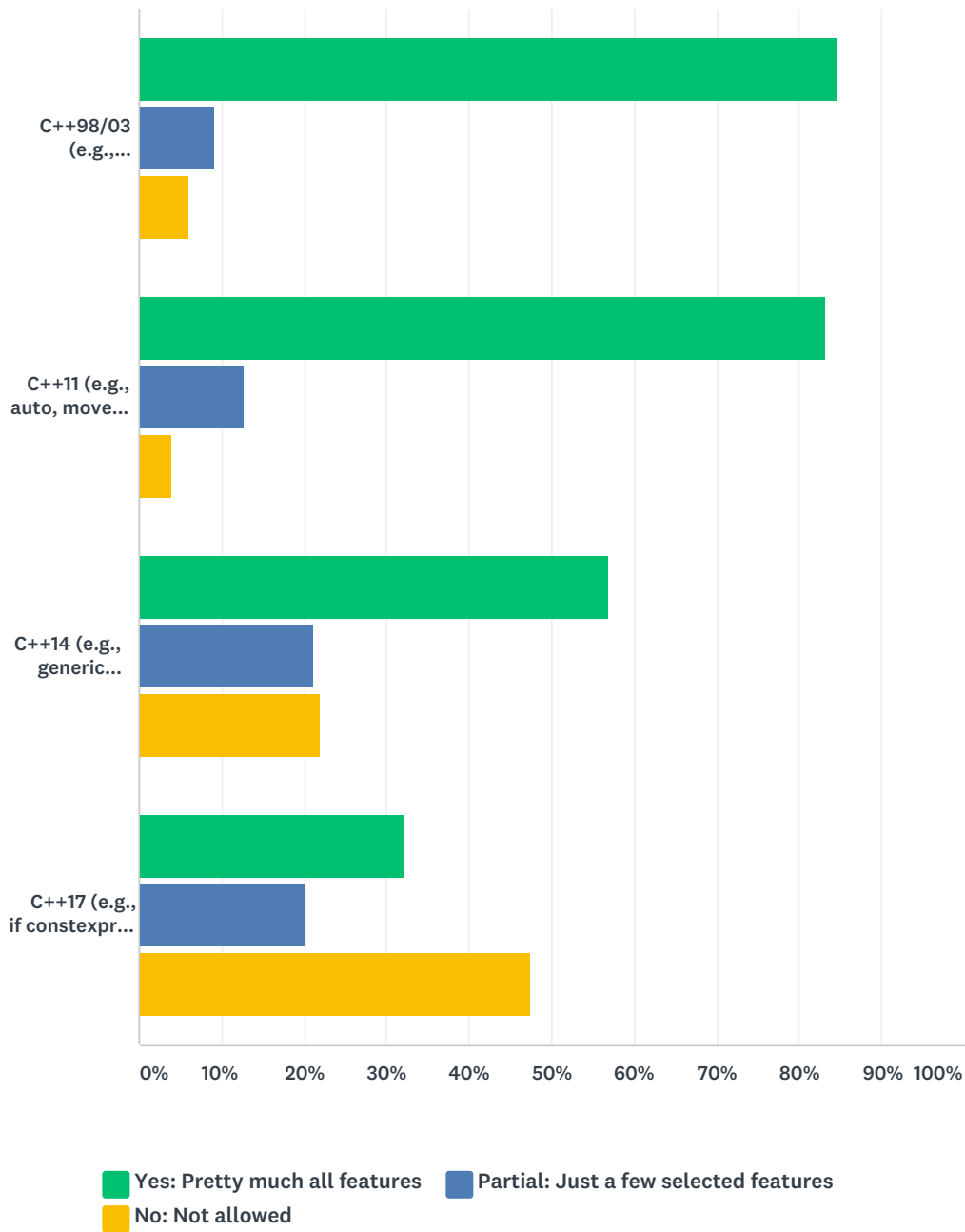


■ Yes: Allowed pretty much anywhere  
■ Partial: Allowed in some parts of the code but not others ■ No: Not allowed

	YES: ALLOWED PRETTY MUCH ANYWHERE	PARTIAL: ALLOWED IN SOME PARTS OF THE CODE BUT NOT OTHERS	NO: NOT ALLOWED	TOTAL	WEIGHTED AVERAGE
Exceptions (throw, try, catch)	47.87% 1,551	32.10% 1,040	20.03% 649	3,240	2.28
Numeric error codes (e.g., errc, error_code, HRESULT)	61.84% 1,979	26.97% 863	11.19% 358	3,200	2.51
Success/failure result class types (e.g., Boost.Expected, Boost.Outcome)	44.57% 1,366	24.08% 738	31.35% 961	3,065	2.13

## Q8 What version(s) of C++ are you allowed to use on your current project (work or school)?

Answered: 3,257 Skipped: 29



	YES: PRETTY MUCH ALL FEATURES	PARTIAL: JUST A FEW SELECTED FEATURES	NO: NOT ALLOWED	TOTAL	WEIGHTED AVERAGE
C++98/03 (e.g., exceptions, templates, RTTI)	84.77% 2,499	9.16% 270	6.07% 179	2,948	2.79
C++11 (e.g., auto, move semantics, =delete/=default, shared_ptr, lambdas)	83.30% 2,588	12.68% 394	4.02% 125	3,107	2.79

## C++ Developer Survey "Lite": 2018-02

C++14 (e.g., generic lambdas, auto return types, general constexpr functions)	56.87% 1,776	21.20% 662	21.93% 685	3,123	2.35
C++17 (e.g., if constexpr, if/switch scoped variables, structured bindings, string_view, optional/any/variant, Parallel STL)	32.20% 982	20.33% 620	47.48% 1,448	3,050	1.85

Q9 As C++ evolves, do you ever encounter difficulty staying abreast with the latest new standard C++ features generally, or with adopting specific new features in your projects? If yes, please describe what is difficult and how it affects you.

Answered: 1,567 Skipped: 1,719

Constexpr<sub>Edge</sub> References<sub>Resources</sub> Toolchain<sub>Impossible</sub>  
Practices<sub>MSVC</sub> Learn<sub>Nope</sub> Code<sub>Modules</sub> Compiler  
Hard to Understand New Features<sub>Colleagues</sub>  
Standard<sub>Amount</sub> Language<sub>Tools</sub> Difficulty<sub>Evolves</sub>  
New Stuff<sub>Dependencies</sub> Older<sub>Past</sub> Books Difficult to Understand

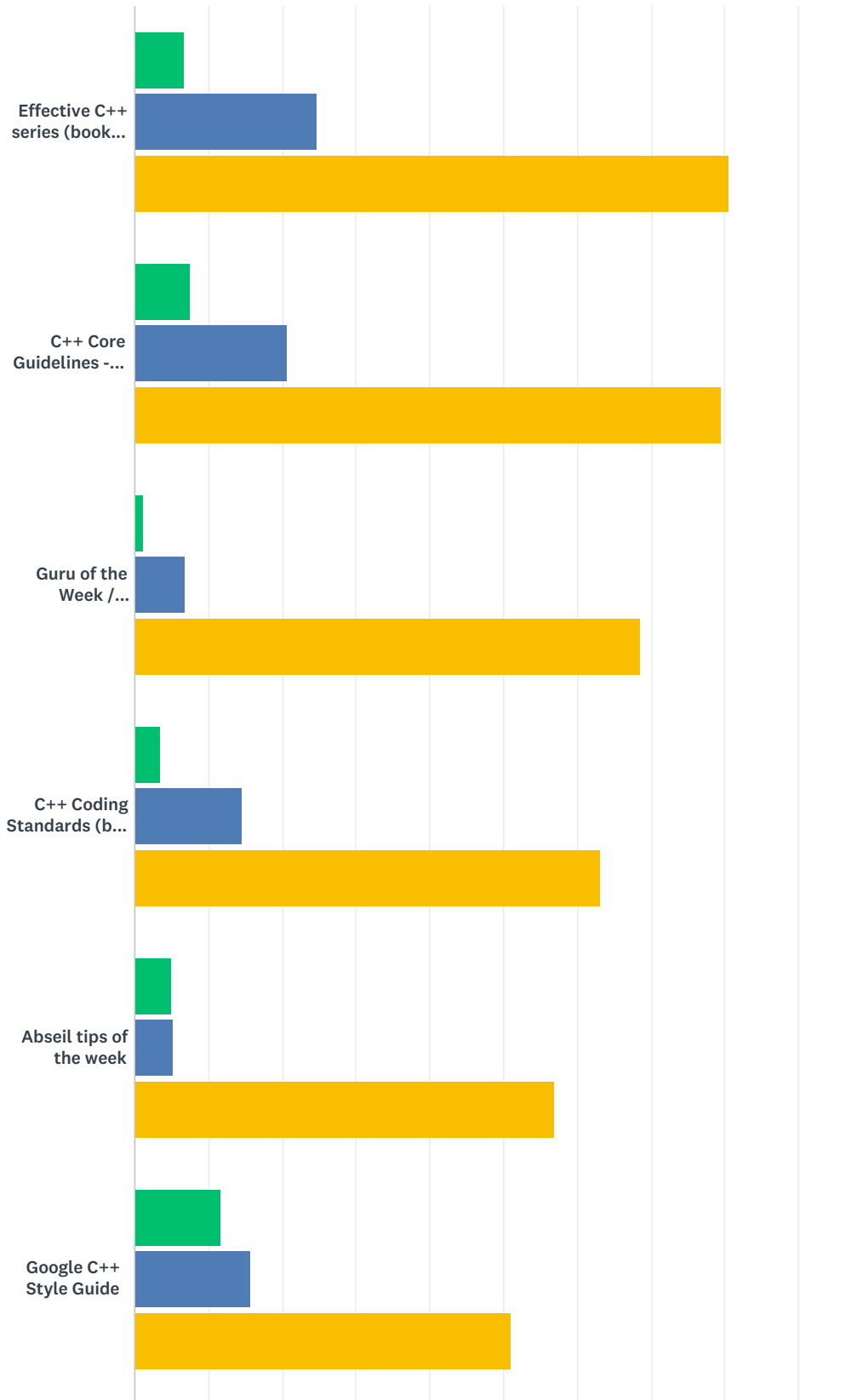
Q10 Can you describe something you need to do regularly when writing C++ code that you wish could be easier in some way? Please elaborate on what makes it difficult, and in what ways making it simpler would help your daily use of C++.

Answered: 1,707 Skipped: 1,579

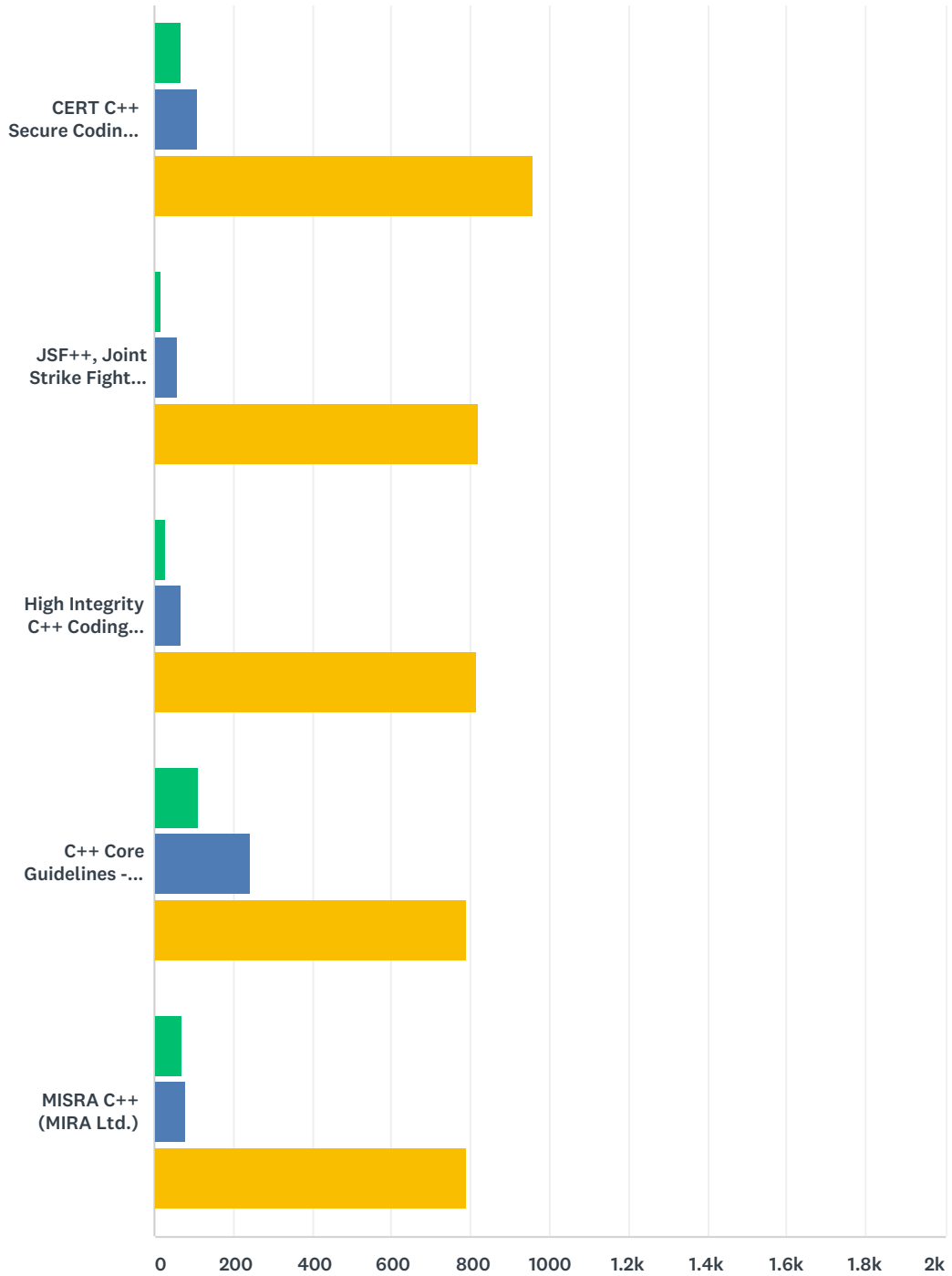
Programming Static Reflection Syntax Overload Resolution Enum Layer  
Iterators Testing Dependency Management  
Frustrating Standard Little Compile Unicode Code  
Template Metaprogramming Library Start Std Faster  
Package Manager Equivalent Concepts Designated Initializers  
Values Typedef Header Files Default

# Q11 Which of these coding advice/guidelines sources do you or your team actively use?

Answered: 2,808 Skipped: 478



## C++ Developer Survey "Lite": 2018-02



- All or part is required by my company/team, with enforcement tools (e.g., linters, ch
- All or part is required by my company/team, but without automated enforcement
- I consult this on my own for guidelines and advice

	ALL OR PART IS REQUIRED BY MY COMPANY/TEAM, WITH ENFORCEMENT TOOLS (E.G., LINTERS, CHECKERS)	ALL OR PART IS REQUIRED BY MY COMPANY/TEAM, BUT WITHOUT AUTOMATED ENFORCEMENT	I CONSULT THIS ON MY OWN FOR GUIDELINES AND ADVICE	TOTAL
Effective C++ series (books by Scott Meyers)	6.03% 135	21.98% 492	71.98% 1,611	2,238

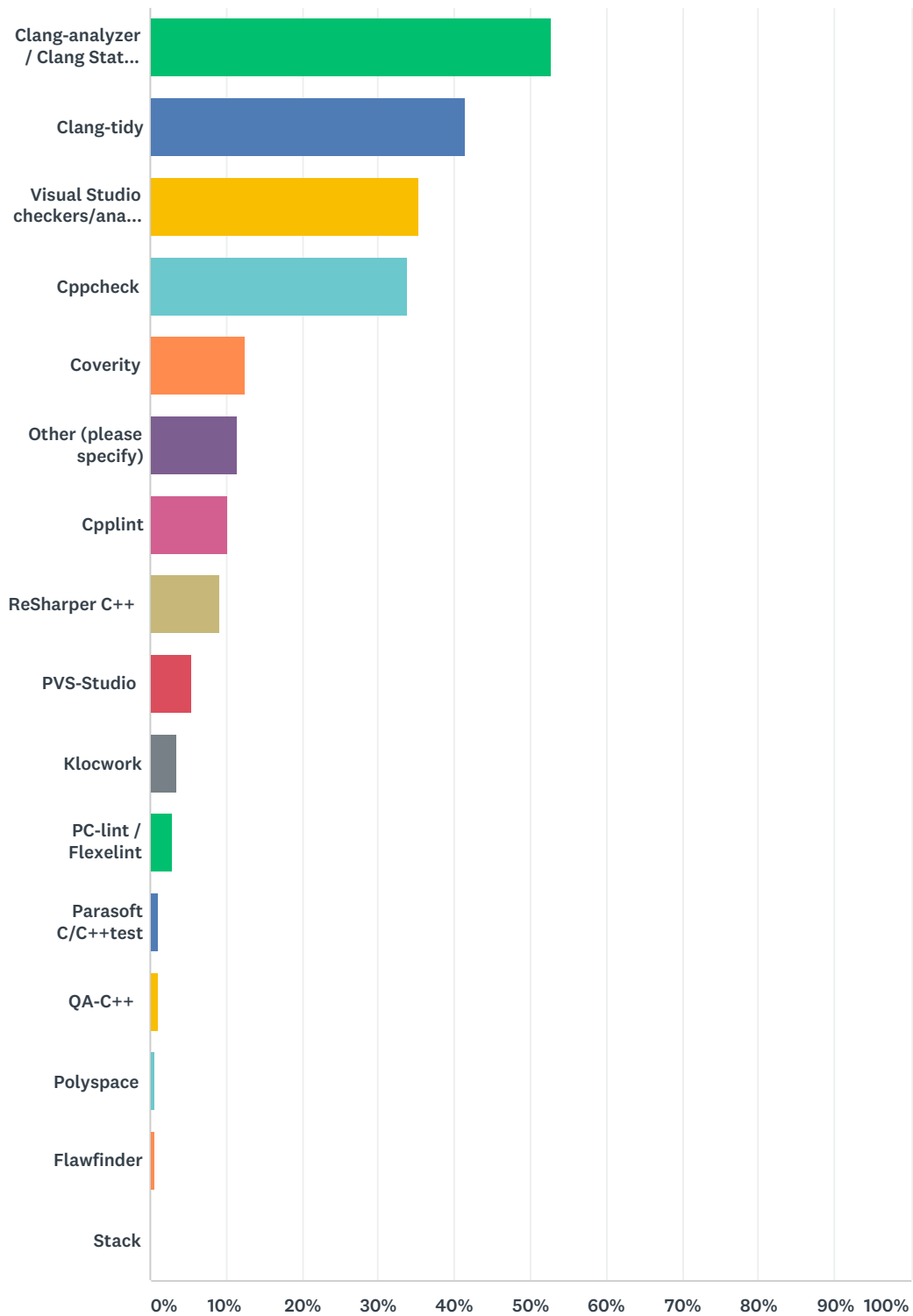


## C++ Developer Survey "Lite": 2018-02

C++ Core Guidelines - main project ( <a href="https://github.com/isocpp/CppCoreGuidelines">github.com/isocpp/CppCoreGuidelines</a> )	6.92% 149	19.17% 413	73.91% 1,592	2,154
Guru of the Week / Exceptional C++ series (blog/books by Herb Sutter)	1.76% 27	8.97% 138	89.27% 1,373	1,538
C++ Coding Standards (book by Herb Sutter and Andrei Alexandrescu)	4.43% 72	17.91% 291	77.66% 1,262	1,625
Abseil tips of the week	7.51% 101	7.81% 105	84.68% 1,139	1,345
Google C++ Style Guide	15.01% 236	19.91% 313	65.08% 1,023	1,572
CERT C++ Secure Coding Standard ( <a href="http://www.securecoding.cert.org">www.securecoding.cert.org</a> )	5.74% 65	9.71% 110	84.55% 958	1,133
JSF++, Joint Strike Fighter Air Vehicle Coding Standards (Lockheed Martin)	2.01% 18	6.47% 58	91.53% 821	897
High Integrity C++ Coding Standard (Programming Research)	3.08% 28	7.16% 65	89.76% 815	908
C++ Core Guidelines - a company- specific fork/branch augmented with internal rules	9.77% 112	21.29% 244	68.94% 790	1,146
MISRA C++ (MIRA Ltd.)	7.45% 70	8.52% 80	84.03% 789	939

## Q12 Which of these tools do you or your team use for guideline enforcement or other code quality/analysis? (select all that apply)

Answered: 2,584 Skipped: 702



ANSWER CHOICES

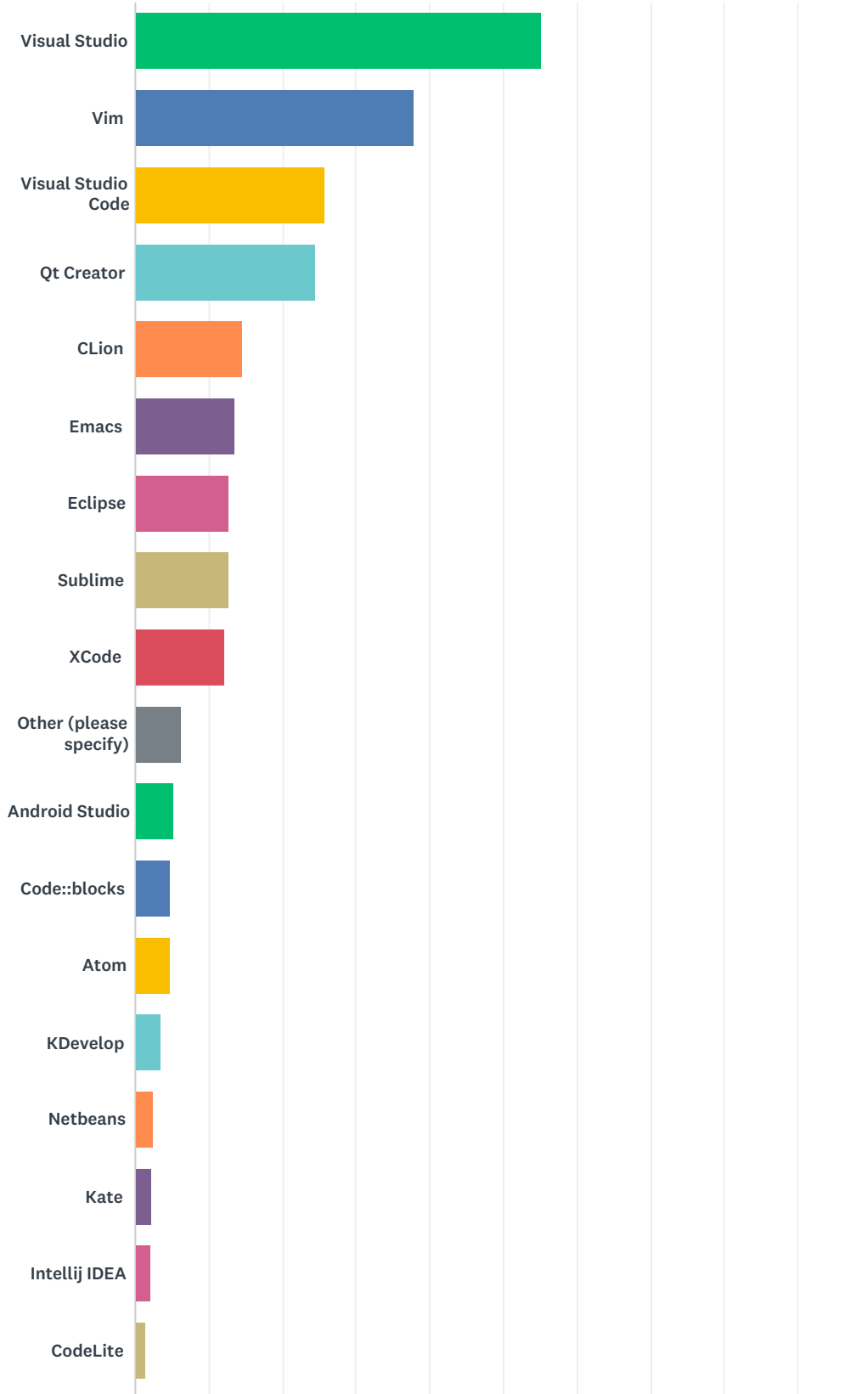
RESPONSES

## C++ Developer Survey "Lite": 2018-02

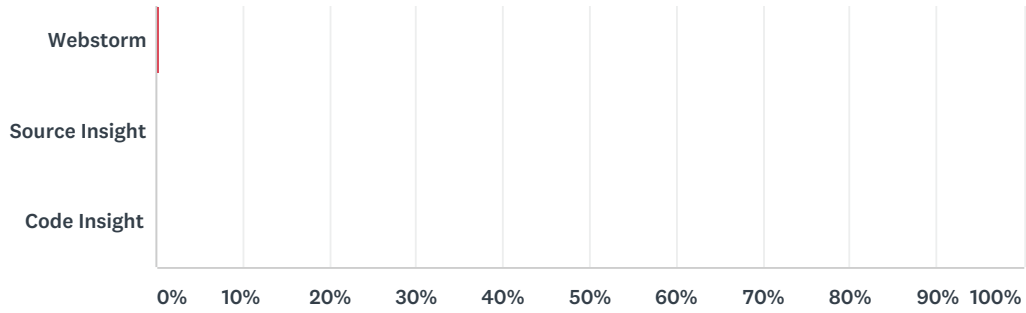
Clang-analyzer / Clang Static Analyzer	52.63%	1,360
Clang-tidy	41.41%	1,070
Visual Studio checkers/analyzers	35.33%	913
Cppcheck	33.94%	877
Coverity	12.50%	323
Other (please specify)	11.42%	295
Cpplint	10.22%	264
ReSharper C++	9.29%	240
PVS-Studio	5.42%	140
Klocwork	3.60%	93
PC-lint / Flexelint	2.90%	75
Parasoft C/C++test	1.12%	29
QA-C++	1.04%	27
Polyspace	0.58%	15
Flawfinder	0.54%	14
Stack	0.31%	8
Total Respondents: 2,584		

# Q13 Which development environments (IDEs) or editors do you use for C++ development?

Answered: 3,240 Skipped: 46



## C++ Developer Survey "Lite": 2018-02



ANSWER CHOICES	RESPONSES	
Visual Studio	55.28%	1,791
Vim	37.93%	1,229
Visual Studio Code	25.77%	835
Qt Creator	24.41%	791
CLion	14.66%	475
Emacs	13.55%	439
Eclipse	12.78%	414
Sublime	12.78%	414
XCode	12.22%	396
Other (please specify)	6.20%	201
Android Studio	5.15%	167
Code::blocks	4.85%	157
Atom	4.72%	153
KDevelop	3.58%	116
Netbeans	2.56%	83
Kate	2.38%	77
Intellij IDEA	2.04%	66
CodeLite	1.54%	50
Webstorm	0.37%	12
Source Insight	0.31%	10
Code Insight	0.00%	0
Total Respondents: 3,240		

Q14 When you last upgraded your IDE or compiler, what were some of the main challenges (e.g., time cost, breaking changes, third party library incompatibility)? Please include which IDE/compiler version you migrated from and to.

Answered: 1,231 Skipped: 2,055

Eclipse Big Issues CLion Binary Compatibility Challenges Configuration  
Support Deprecated Problems Takes Visual Studio  
Vim **Compiler** Build System Gcc Emacs Party Library  
Trying VS2017 macOS Breaking Changes Syntax Highlighting  
Platforms Linker Errors Qt Creator

Q15 If you could wave a magic wand and change one thing about any part of C++, what would it be, and how would that change help your daily work?

Answered: 1,775 Skipped: 1,511

Reduce Unit Test Package Manager Msvc Syntax Faster Build Default  
Better Support Language Pointer Library Rust Code Implicit  
Compile Header Files Standard Metaclasses Template  
Drop Remove Size Std Move Semantics Compatibility ABI

## Q16 Do you have any additional feedback for C++ standardization?

Answered: 1,117 Skipped: 2,169

Rust Start Removing Stuff Meta Classes ABI Great Work Reason Job  
Continue Modules Good Idea Library Simpler Language  
Feedback Standard Awesome Features Experience  
Good Work Deprecate Code Efficient Stop Moving Forward  
Package Manager Easier to Learn Hard Work String