SMARTBEAR

The State of Code Quality 2016:

Trends & Insight into Dev Collaboration

Collaborator

Preface

This survey was designed to establish benchmarks for the software industry on how organizations are developing and maintaining the quality of software in 2016. The following report will cover:

- Perceptions on Code Quality
- Approaches to Code Review
- Code Review Tools and Decision Making
- Software Development Tools

Methodology

SmartBear Software conducted a global online survey over the course of two weeks during the month of December 2015. The findings presented are based upon the aggregated responses from over 600 software developers, testers, IT/operations professionals, and business leaders representing more than 30 different industries. Participants in the survey work on software teams ranging from less than five employees up to more than 50 employees, and work for companies ranging from small businesses, with less than 25 employees, to enterprise organizations, with 10,000 employees or more.

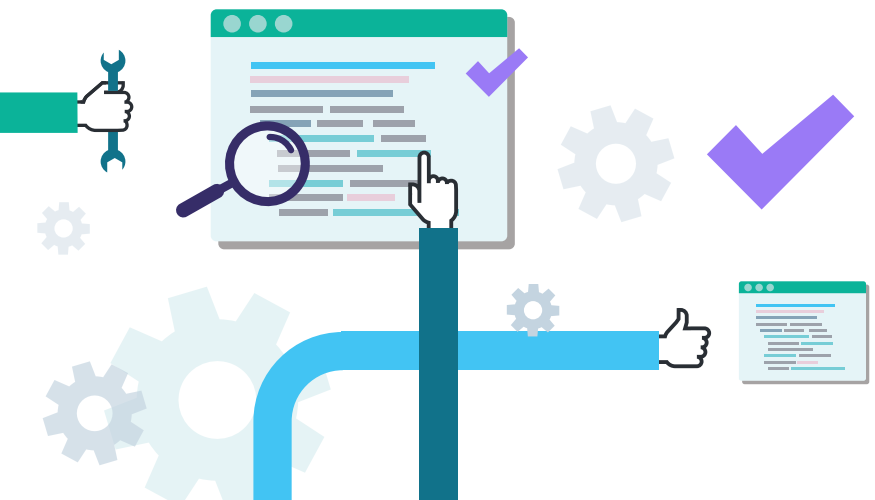
Content

Key Findings

1

Two-thirds of respondents are satisfied with the quality of software they help build

- Two-thirds of respondents said they either agree (**54.5%**) or strongly agree (**12.3%**) that they are satisfied with their software quality.
- **10%** of respondents either disagree (**8.8%**) or strongly disagree (**1.1%**) that they are satisfied with their software quality.



Code review is looked at as the #1 way to improve code quality

2

When it comes to code quality, code review is king

- Code review is looked at as the **#1 way** to improve code quality.
- **90%** of respondents say that improved code quality is the biggest benefit of code review.
- **69%** of respondents say that improving code quality is the biggest business driver determining the need for a code quality tool.

3

Nearly two-thirds of respondents are doing code review, but for most it is still an ad-hoc process

- **62.7%** of respondents are doing code review.
- **71.9%** are doing ad-hoc, or "over the shoulder" code review.
- **63.4%** are doing tool-based code review.
- **52.8%** are doing meeting-based code review.

4

Workload, time constraints, and manpower are the biggest obstacles to code review

- **65.6%** of respondents say workload is the biggest obstacle preventing them from doing the level of code review they desire.
- **43.7%** say deadlines/time constraints is the biggest obstacle.
- **39.4%** say that lack of manpower is the biggest obstacle.

5

Using a tool for code review enables teams to review code on a more frequent basis

- **51%** of respondents are using at least one tool for code review.
- **23.6%** of respondents that are doing tool-based code review are doing it on a daily basis; **20.6%** are doing it on a weekly basis.
- **3.5%** of respondents that are doing meeting-based code review are doing it on a daily basis; **17.7%** are doing it on a weekly basis.
- **14.2%** of respondents that are doing ad-hoc code review are doing it on a daily basis; **29.1%** are doing it on a weekly basis.

6

Git and Subversion are the most commonly used software configuration management tools (SCMs)

- **43.4%** of respondents are using Git as their SCM.
- **27%** of respondents are using GitHub or GitHub Enterprise for repository management.
- **One-third** of respondents are using Subversion as their SCM.

7

JIRA outpaces all others for bug tracking and requirements management

- **41.8%** of respondents use Jira for bug tracking.
- **34.2%** of respondents use Jira for requirements management.

Section 1: Code Quality in 2016

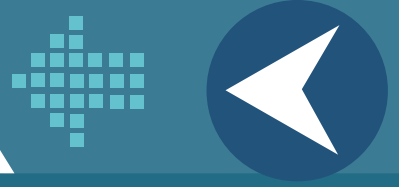
It shouldn't come as a surprise that code quality is a top concern for organizations of all shapes and sizes. But are these organizations meeting the high code standards they set for themselves? How are they ensuring that they maintain the quality of their software? In the first section of the *State of Code Quality 2016* report, we look at how organizations are maintaining code quality in 2016.

Highlights

01 Two-thirds of respondents agree that they are satisfied with the quality of software they help build.

02 More than two-thirds of respondents say they are regularly able to get releases out on time.

03 Code review is looked at as the #1 way to improve code quality.



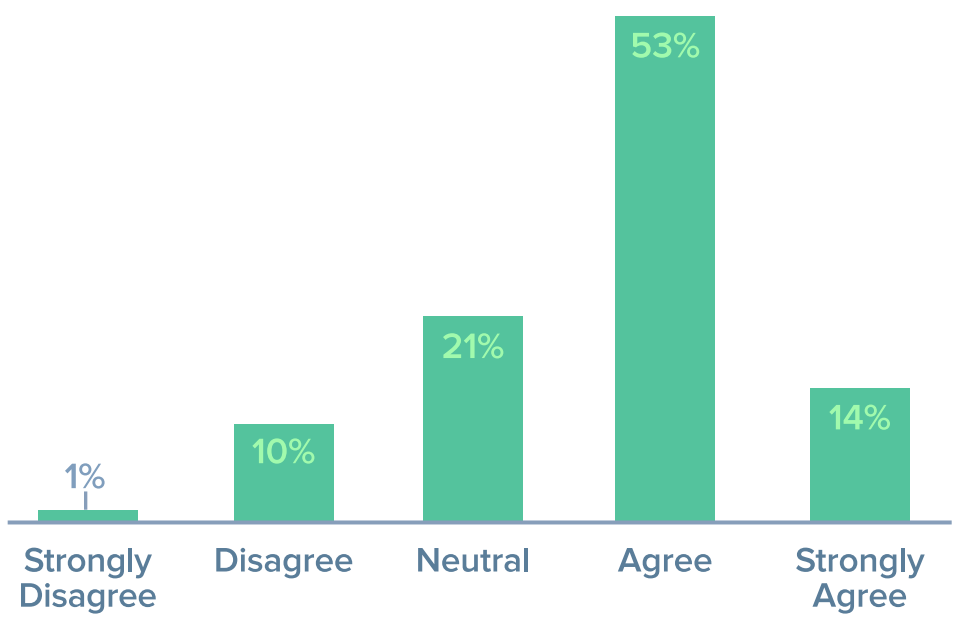
Two-thirds of respondents agree that they are satisfied with the quality of software they help build

Overall, we found that respondents felt positively about the quality of the software they help build. Of those who didn't either agree or strongly agree, just **10%** said they felt negatively about the software they help develop.

The overall positive sentiment was felt across industries and roles within software teams. One area where negative sentiment was higher was within software teams of **50** people or more.

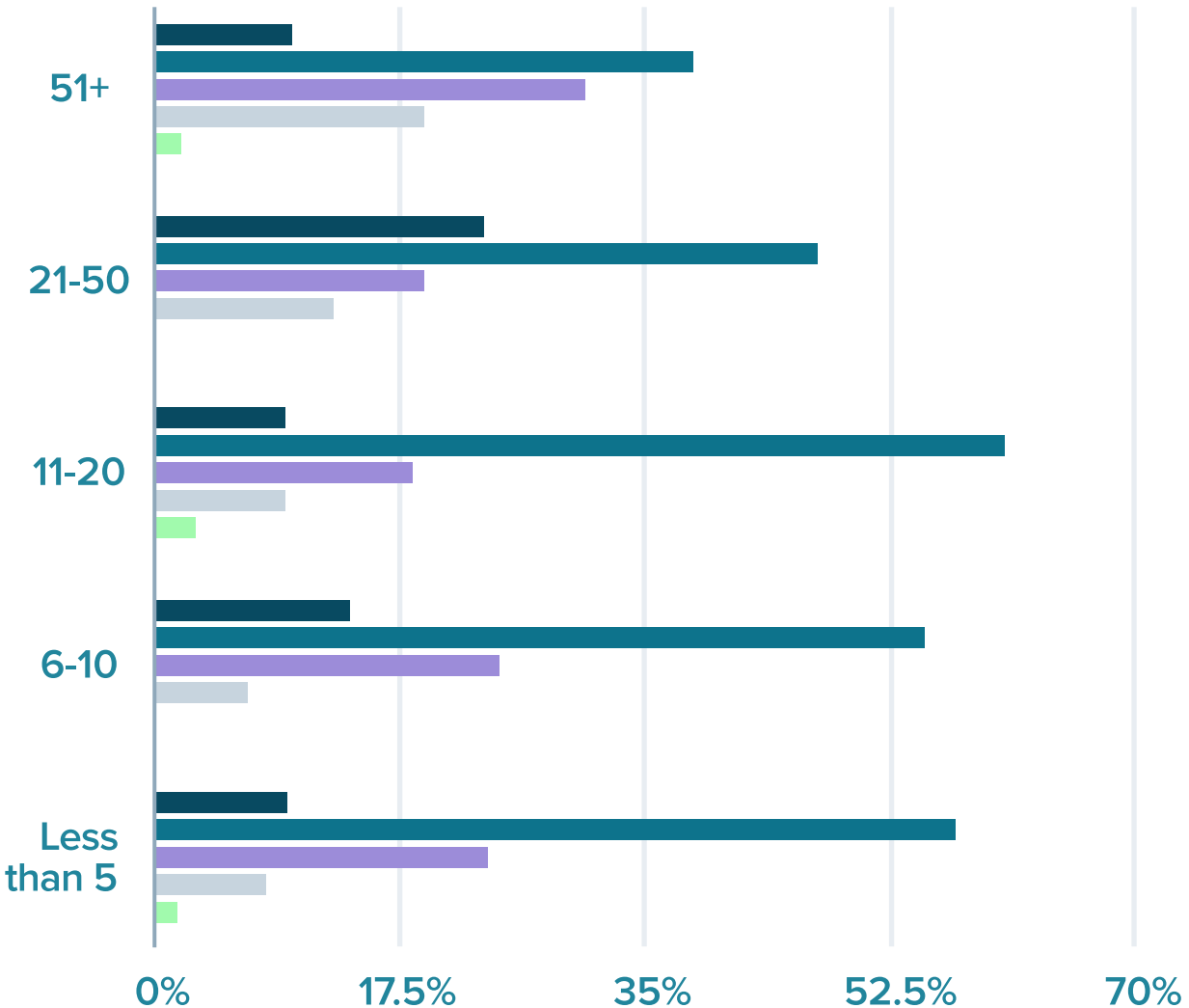
I am satisfied with the overall quality of the software I help deliver (specifically regarding performance, bugs, etc.).

n = 537



1 in 5 respondents on development teams of 50+ said they either disagreed or strongly disagreed that they were satisfied with the quality of code they help build.

by Team Size



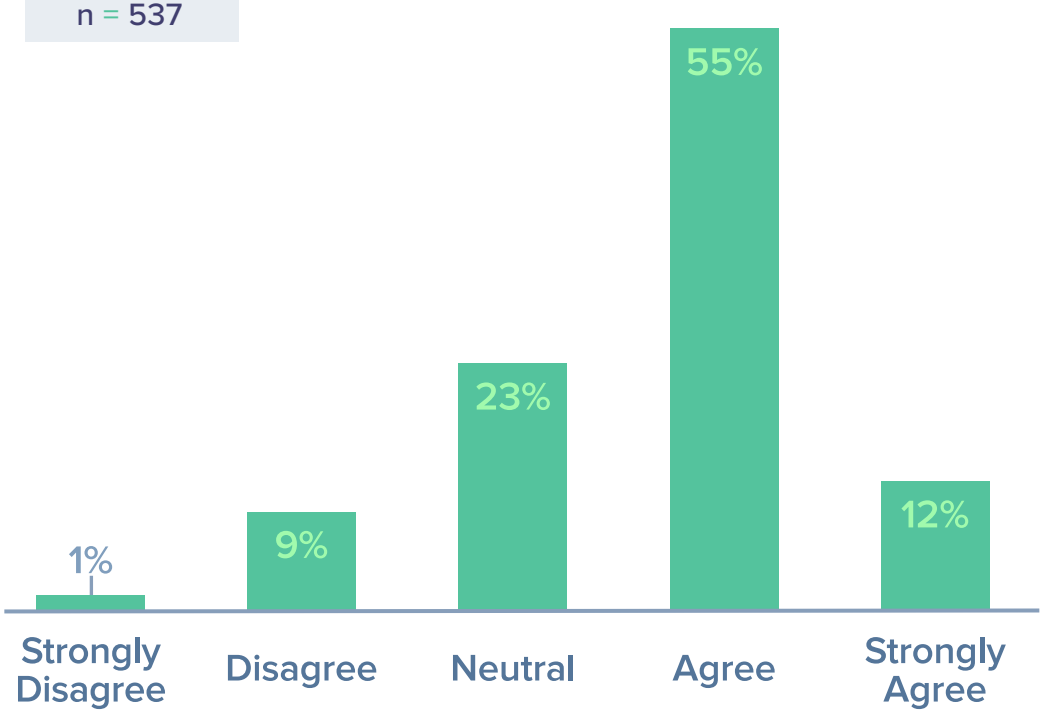
I am satisfied with the overall quality of the software I help deliver (specifically regarding performance, bugs, etc.).

n = 537

More than two-thirds of respondents say they are regularly able to get releases out on time.

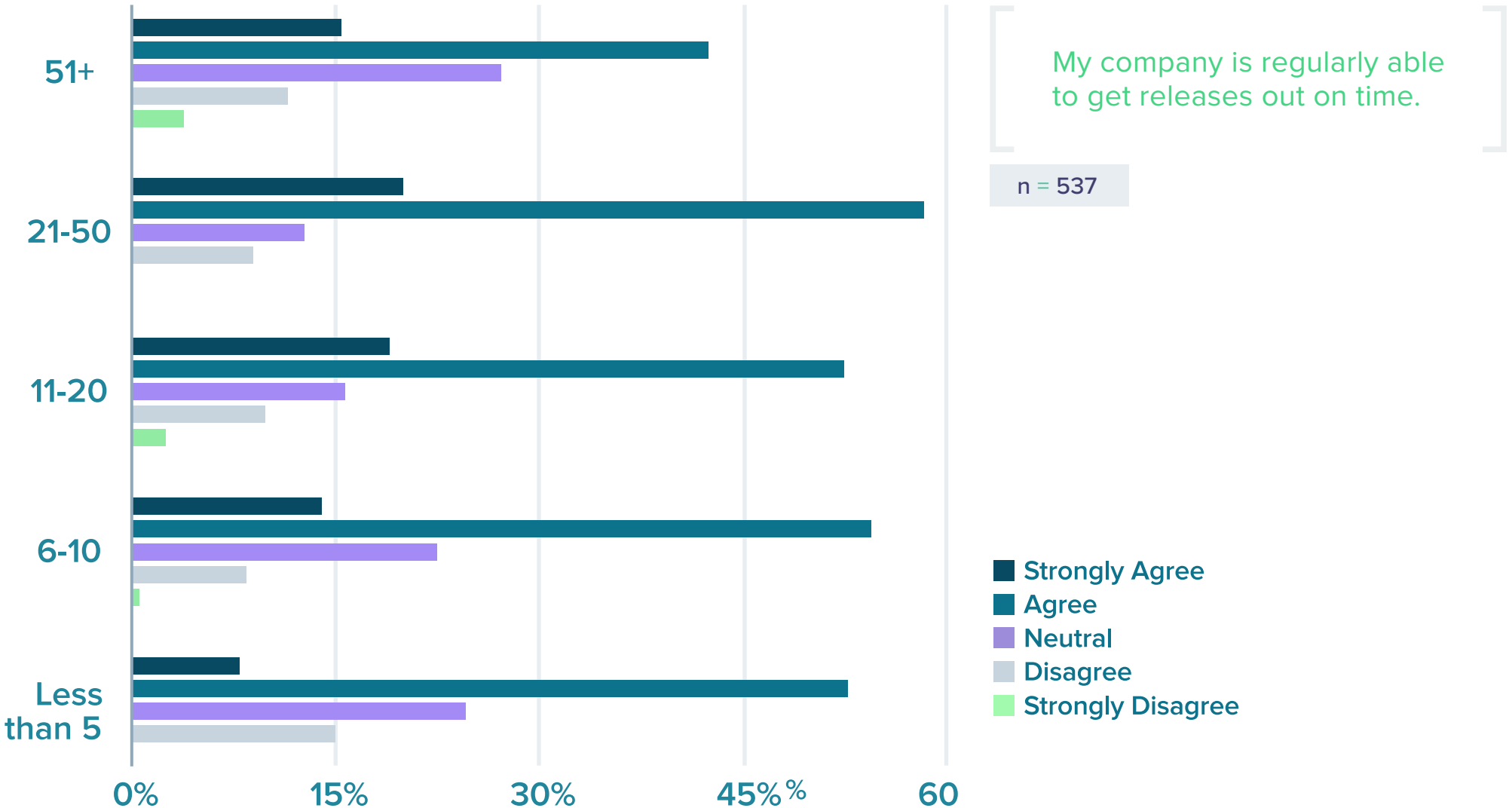
My company is regularly able to get releases out on time.

n = 537



Development teams with 21-50 members were the most positive about their ability to get releases out on time with 78.2% reporting that they either agree or strongly agree.

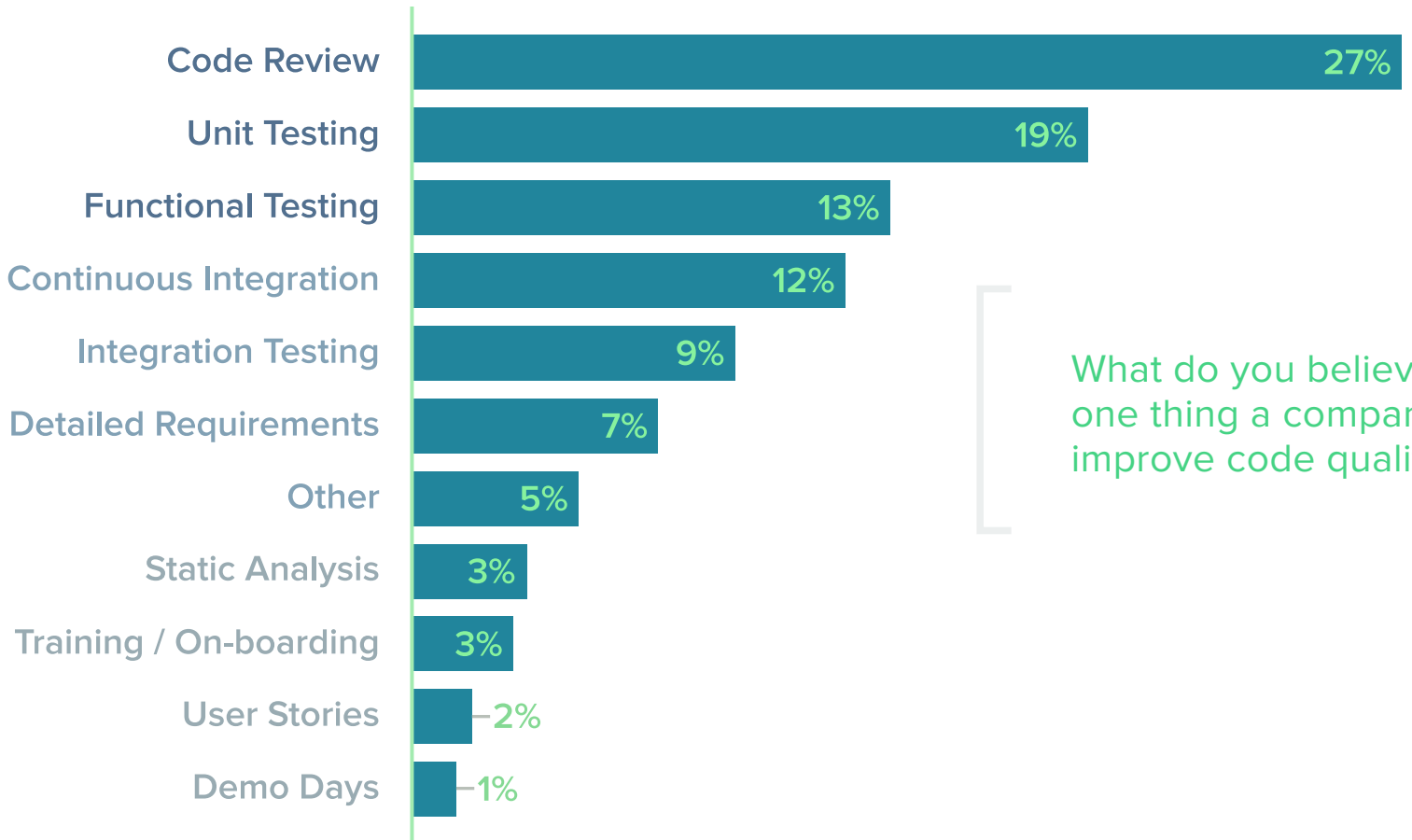
by Team Size



Code review is looked at as the #1 way to improve code quality

When asked, “What do you believe is the number one thing a company can do to improve code quality?” 27.4% of respondents said that code review is the best thing their team should be doing.

Testing is also considered to be a critical method for improving code quality — 18.7% say unit testing and 13% say functional testing are both important for improving code quality. 30.3% of respondents working in organizations with 1,000+ employees agree that code review is the best tool for improving code quality.



What do you believe is the number one thing a company can do to improve code quality?

n = 508

Section 2: Approaches to Code Review

Code review is looked at as the number one method for improving code quality, but the ways in which teams implement code review can often vary. In the second section of the *State of Code Quality 2016* report, we look at how teams are using code review and the challenges that can often get in the way.

Highlights

01 Improved software quality, sharing knowledge across teams, and enhanced maintainability of code are viewed as top benefits of code review.

02 72% of respondents are doing "over the shoulder" code review; 62% are doing it at least once a month.

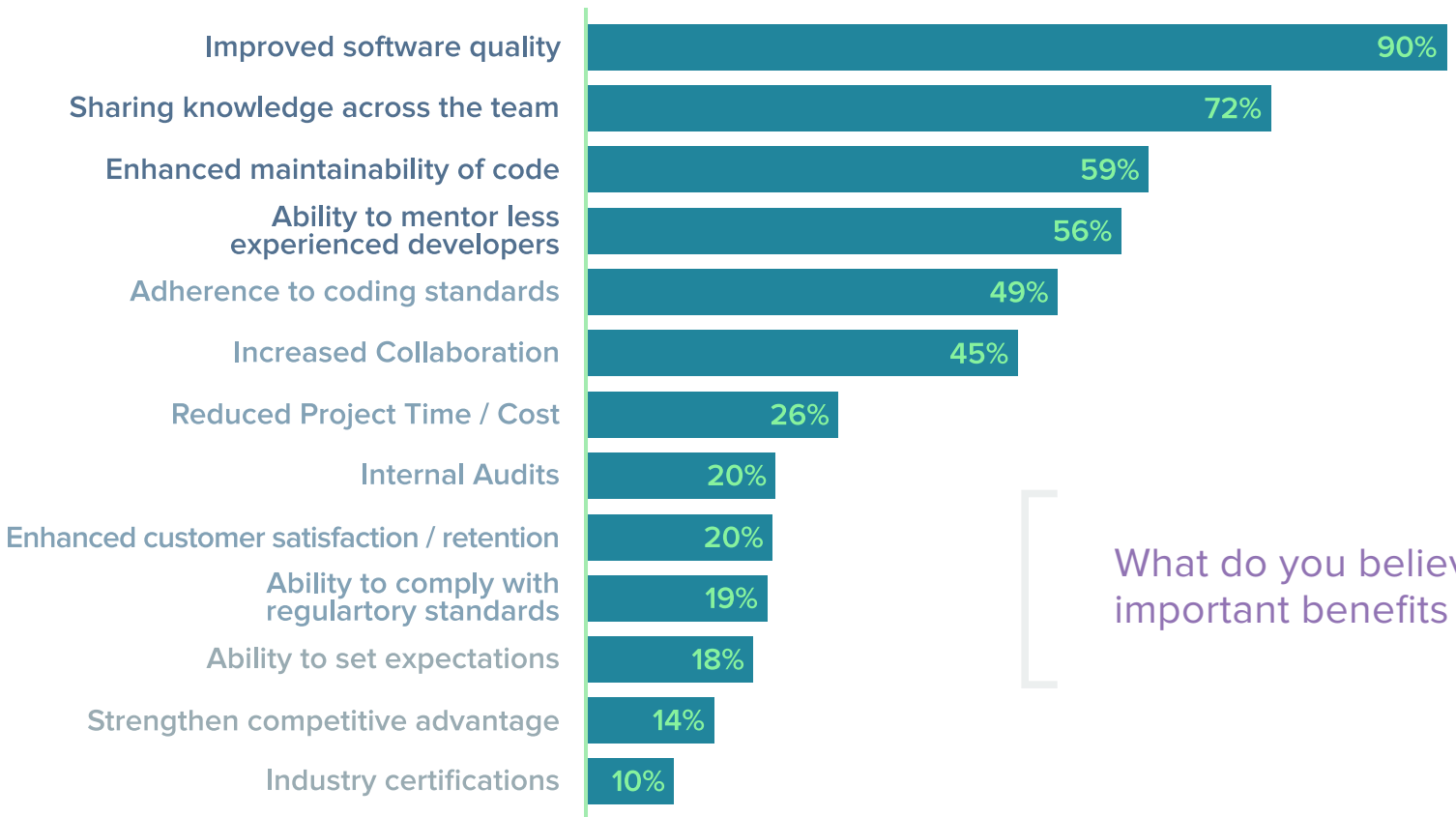
03 Half of respondents are doing meeting-based code review; 37% are doing it at least once a month.

04 63% of respondents are doing tool-based code review; 23.6% are doing it on a daily basis.

05 Workload, time constraints, and lack of manpower are the biggest obstacles to code review.

Improved software quality, sharing knowledge across teams, and enhanced maintainability of code are looked at as top benefits of code review

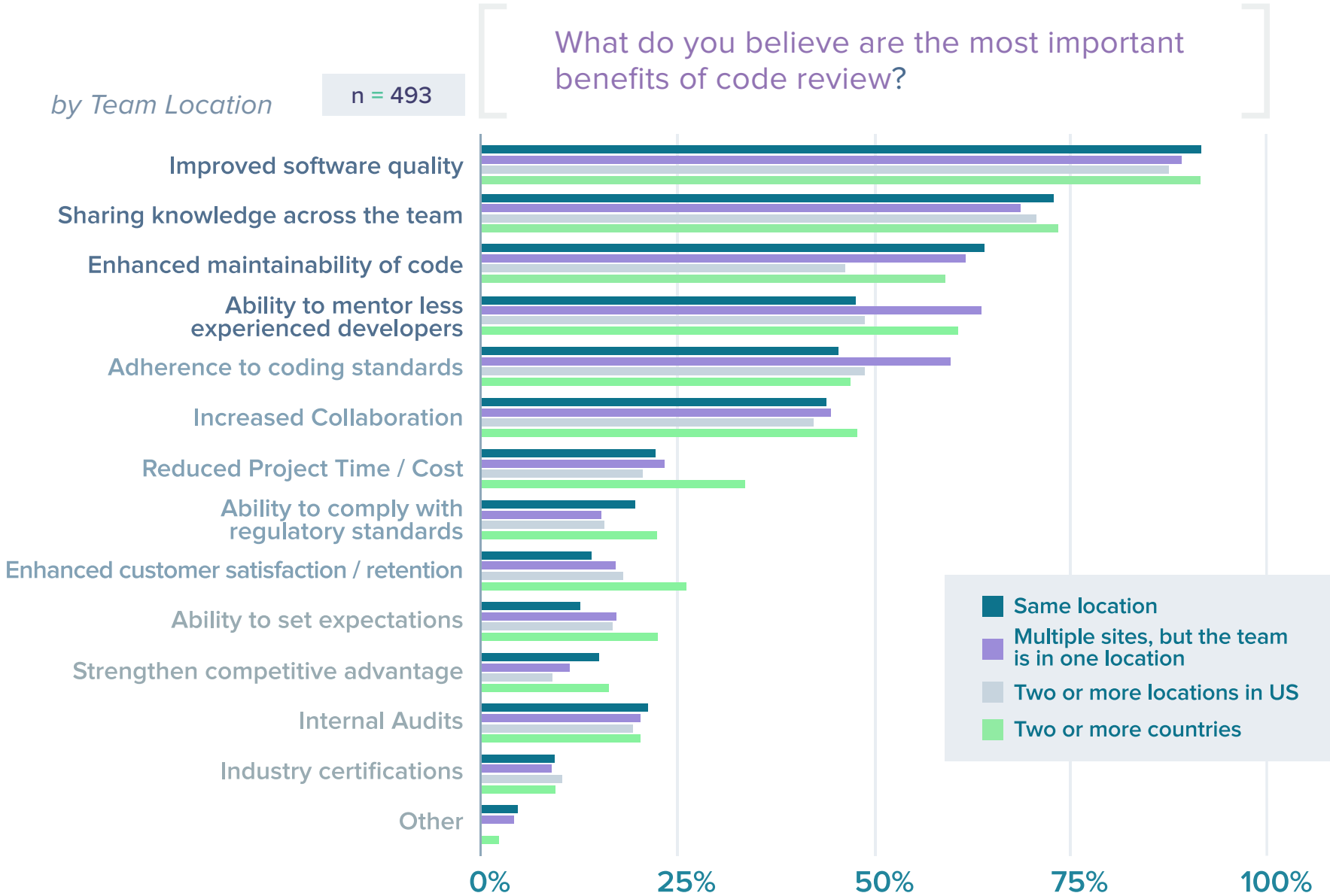
Improved software quality is the number one benefit of doing code review, with 90% of respondents saying that code quality is the best reason for doing code review. Beyond the quality of code, reviews are also viewed as a primary method for improving the ways teams work together, including sharing knowledge across the team, ability to mentor less experienced developers, and increased collaboration.



What do you believe are the most important benefits of code review?

n = 493

One-third of companies with development teams located across two or more countries say that the ability to reduce cost is one of the biggest benefits of code review.



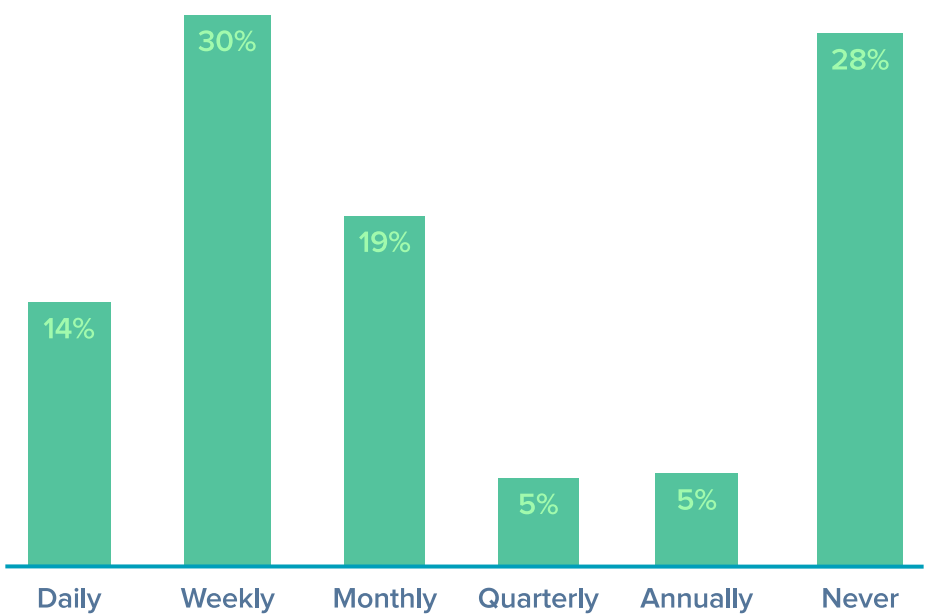
72% of respondents are doing “over the shoulder” code review; 62% are doing it at least once a month

Ad-hoc, or “over the shoulder” code review is the most commonly used type of code review within organizations, with nearly three-quarters of respondents participating in ad-hoc reviews throughout the year.

The majority of respondents who are doing ad-hoc code review do it at least once per month, with a majority of respondents conducting ad-hoc code review on a weekly basis.

How often do you participate in an ad-hoc (“over the shoulder”) code-review process?

n = 508

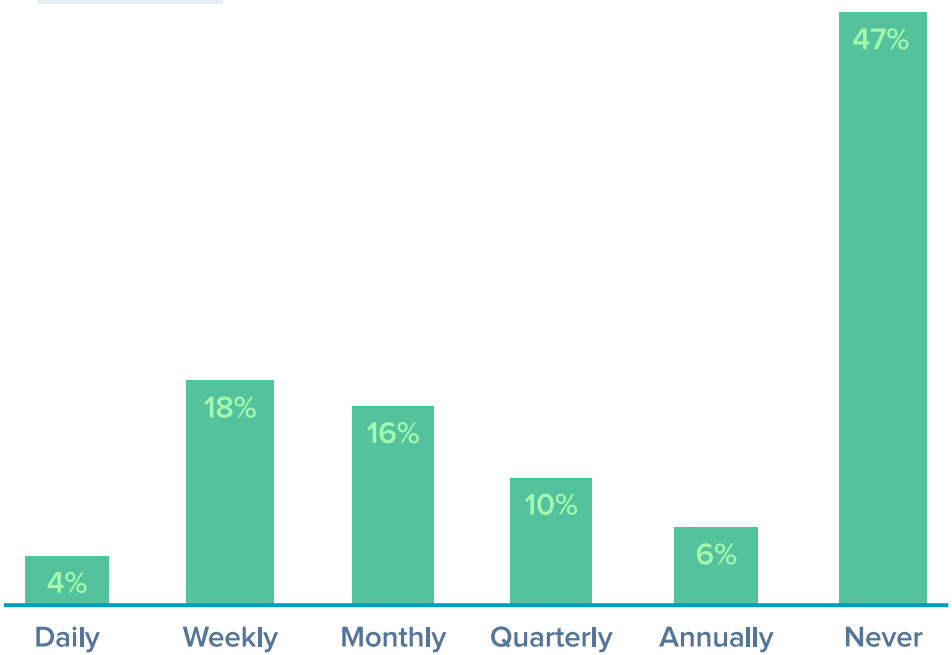


Half of respondents are doing meeting-based code review; 37% are doing it at least once a month

Meeting-based code review is the least commonly used method for code review, with just over half of organizations participating in meeting-based reviews.

How often do you participate in a **meeting-based** code-review process?

n = 508

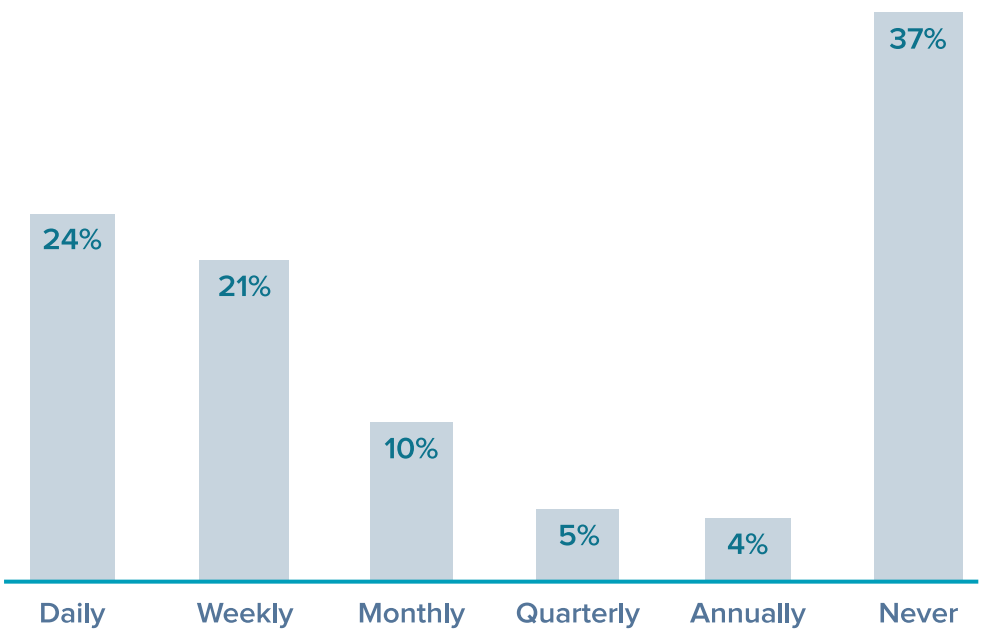


63% of respondents are doing tool-based code review; 23.6% are doing it on a daily basis

Nearly two-thirds of individuals surveyed say that their organization is using a tool to help with code review. Based on survey results, organizations that are doing tool-based code review are doing it on a more consistent basis.

How often do you participate in a **tool-based** code-review process?

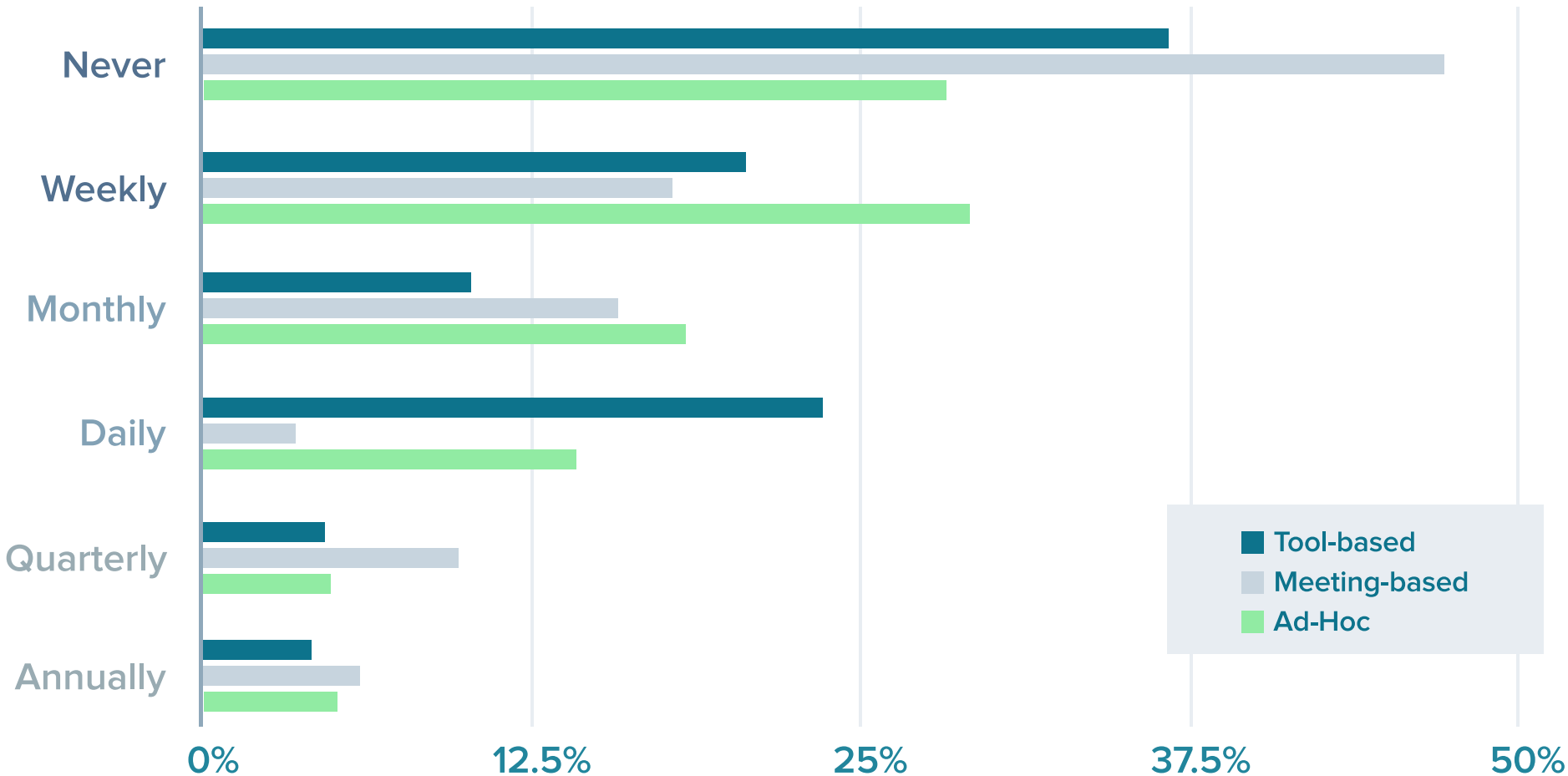
n = 508



23.6% of organizations are doing tool-based code review on a daily basis, compared to 14% for meeting-based code review and less than 5% for ad-hoc code review.

How often do you participate in a ad-hoc, meeting-based, or tool-based code-review process?

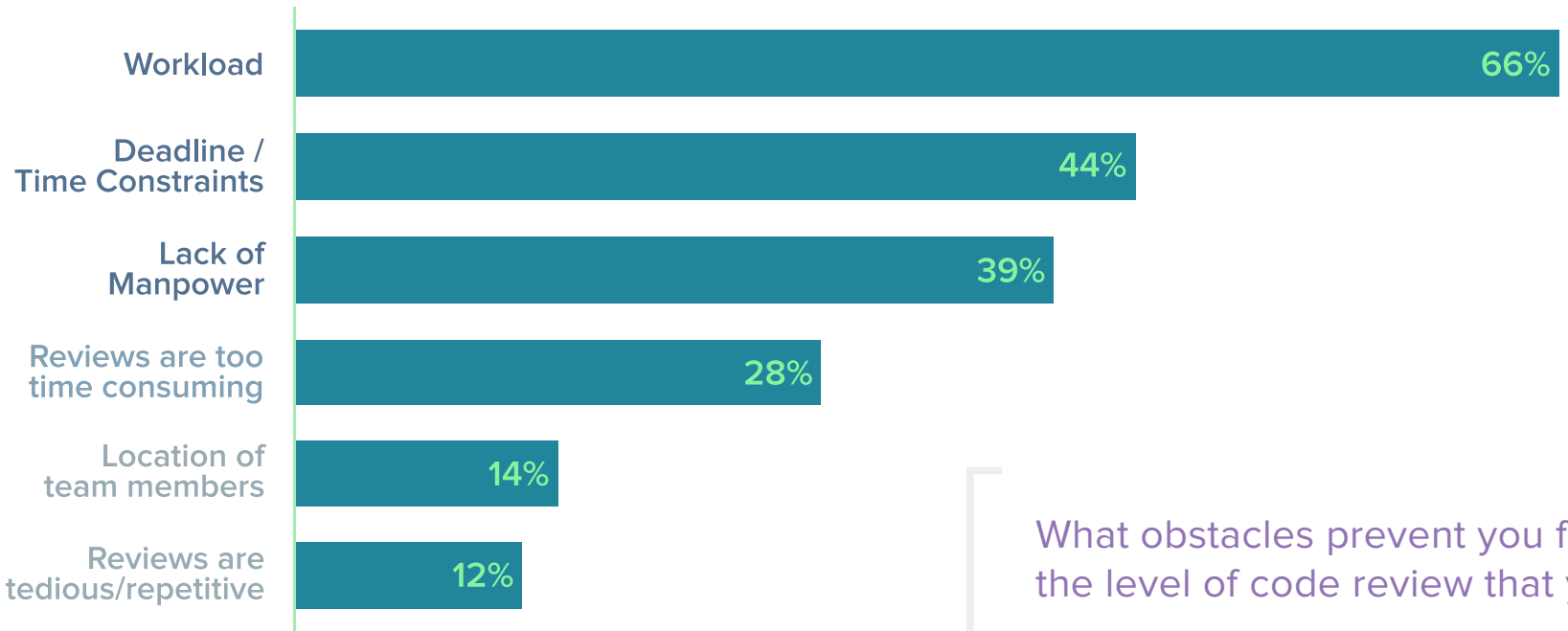
n = 508



Workload, time constraints, and manpower are the biggest obstacles to code review

While teams understand the importance of code review, they often face obstacles when it comes to implementing it into their team or organization. Nearly two-thirds of respondents say that workload is the number one obstacle facing their team when it comes to doing code review.

Similarly, time restraints and a lack of manpower are also looked at as primary roadblocks as teams struggle to allocate the necessary resources to conduct code reviews.



n = 508

What obstacles prevent you from doing the level of code review that you desire?

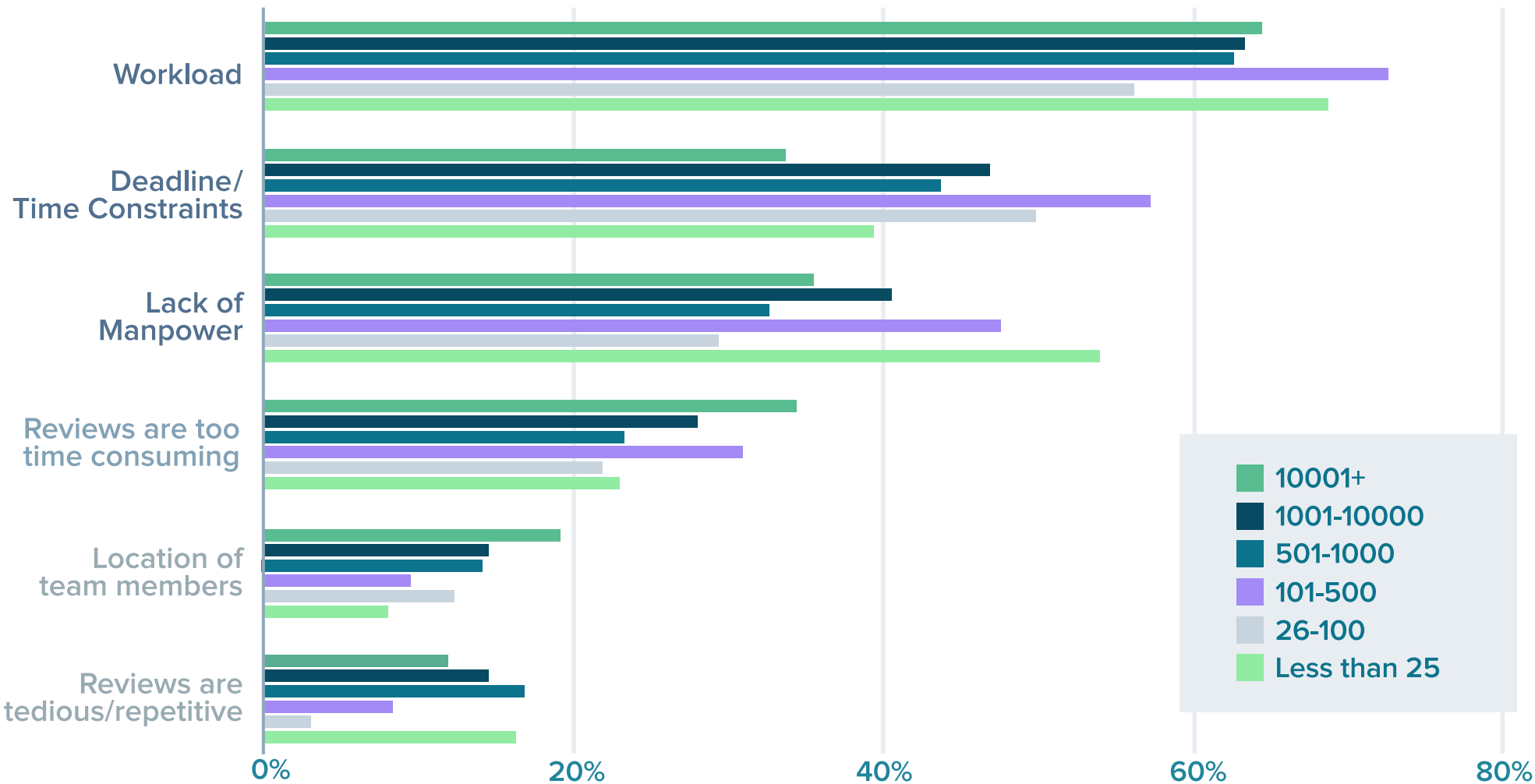
Workload is the #1 challenge for organizations of all sizes

More than half of respondents from organizations with less than 25 employees also cited lack of manpower as their biggest obstacle.

What obstacles prevent you from doing the level of code review that you desire?

by Company Size

n = 508



Section 3: Code Review Tools

Organizations — both large and small — can benefit from using a tool to assist with code review. There are a number of business drivers that determine the need for a code review tool, and choosing the right tool will depend on the needs of your software team. In section 3 of the *State of Code Quality* report, we take a closer look at how teams are choosing code review tools and the factors that go into selecting a tool.

Highlights

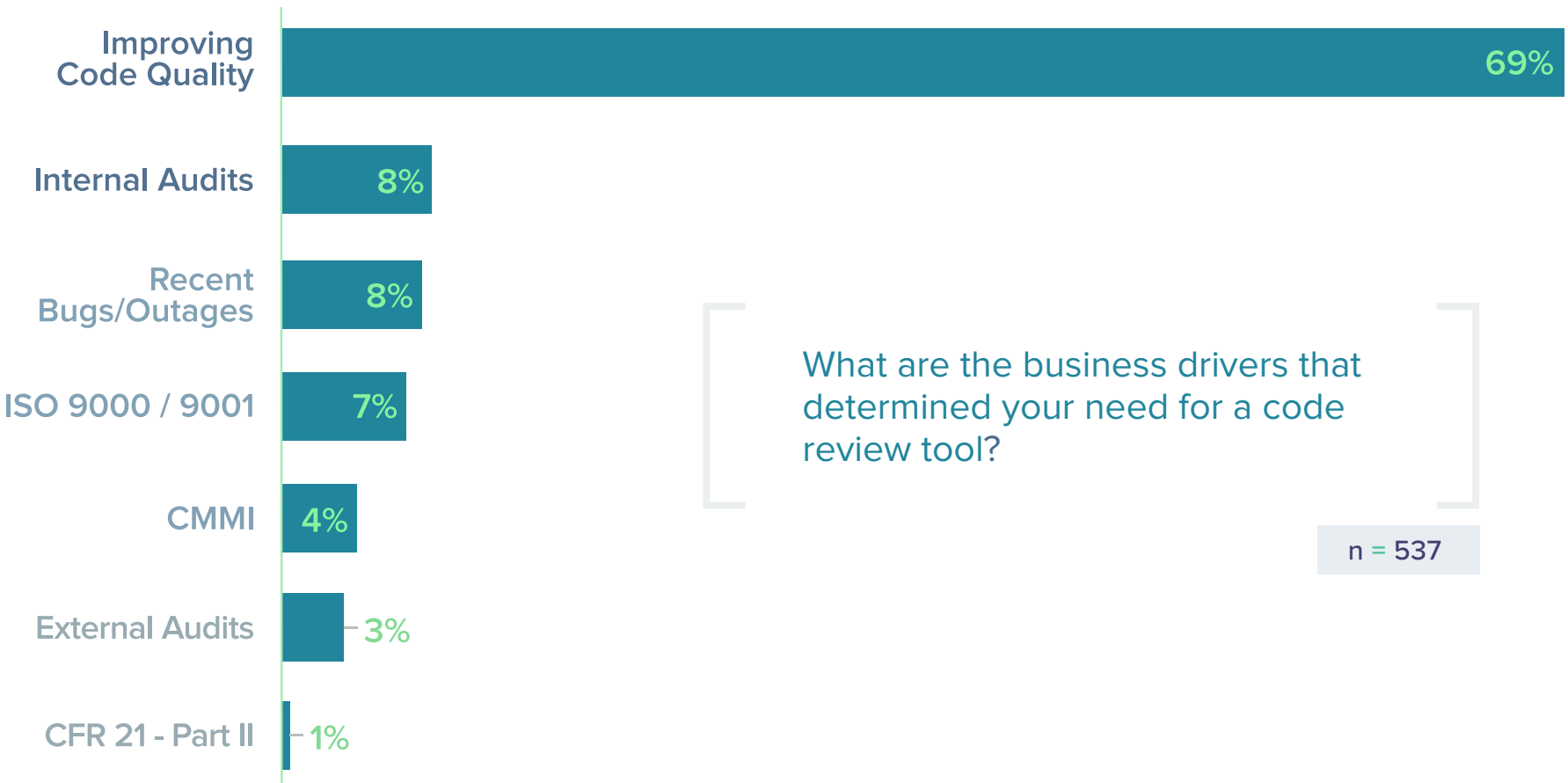
01 Improving code quality is the number one reason respondents use to determine the need for a code review tool.

02 51% of respondents are using at least one code review tool; 1 in 5 are using Collaborator.

03 Workload, budget, and time constraints are the biggest obstacles to doing tool-assisted code review.

Improving code quality is the top reason for using a code review tool

An overwhelming majority of respondents say that “improving code quality” is the number one business driver that determines their need for a code review tool. Improved code quality outnumbered all other factors for using a code review tool combined, by a ratio of 3:1. Along with improved code quality, finding bugs and product outages, as well as audits — both internal and external — were driving factors organizations use to determine the need for a code review tool.



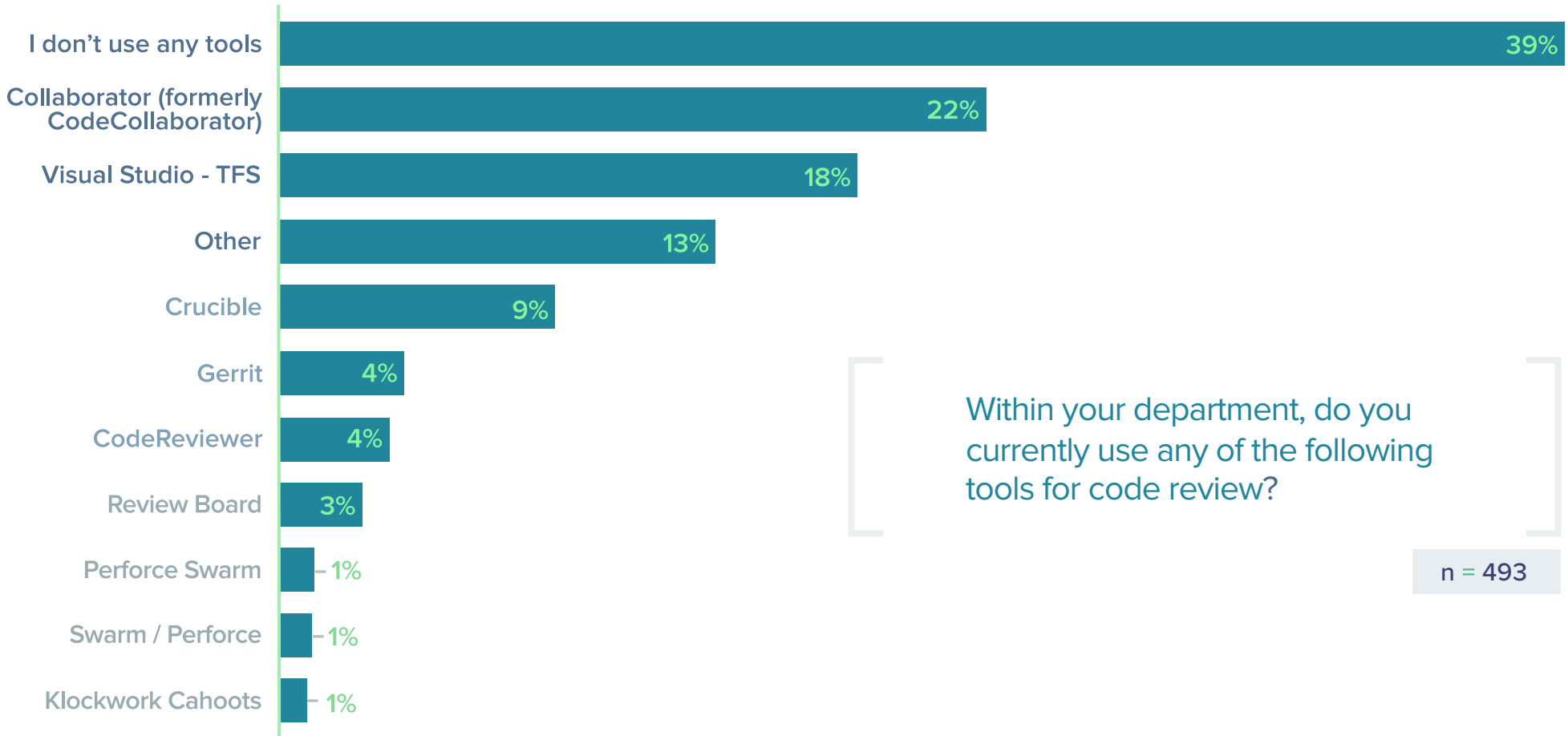
What are the business drivers that determined your need for a code review tool?

n = 537

Nearly 51% of respondents are using at least one tool for code review; 1 in 5 are using Collaborator

The *State of Code Quality 2016* report is an industry-wide survey that was not limited to SmartBear customers. We are proud to see that Collaborator, our code review tool, was the most commonly used tool among organizations that are doing tool-assisted code review.

In addition, we also found that nearly 40% of organizations are not using a tool for code reviews.



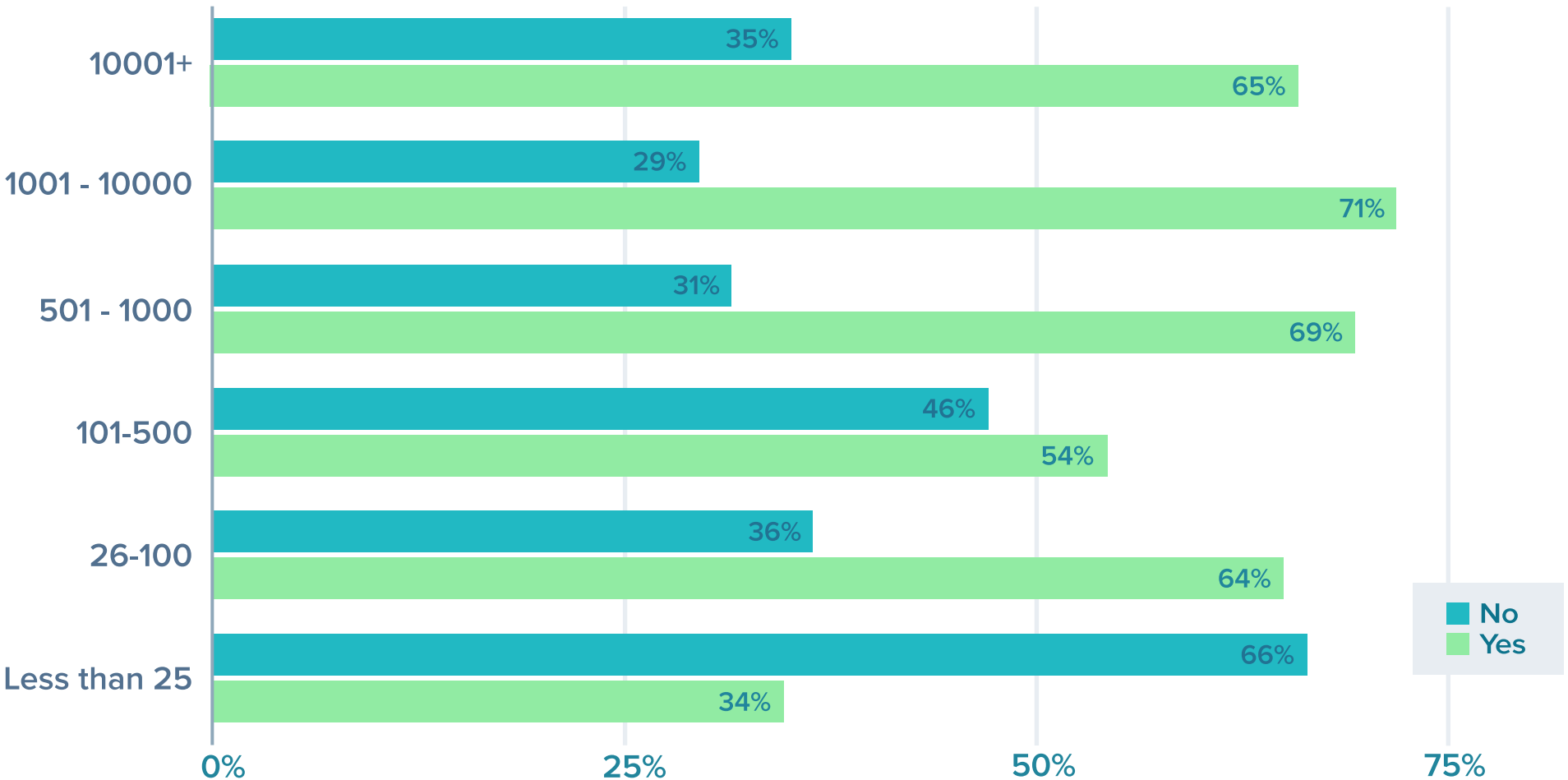
Within your department, do you currently use any of the following tools for code review?

n = 493

Two-thirds of organizations with 1,000+ employees are using a code review tool; 1 in 3 companies with 25 or less employees are using a tool for code review

Code Review Tool Usage by Company Size

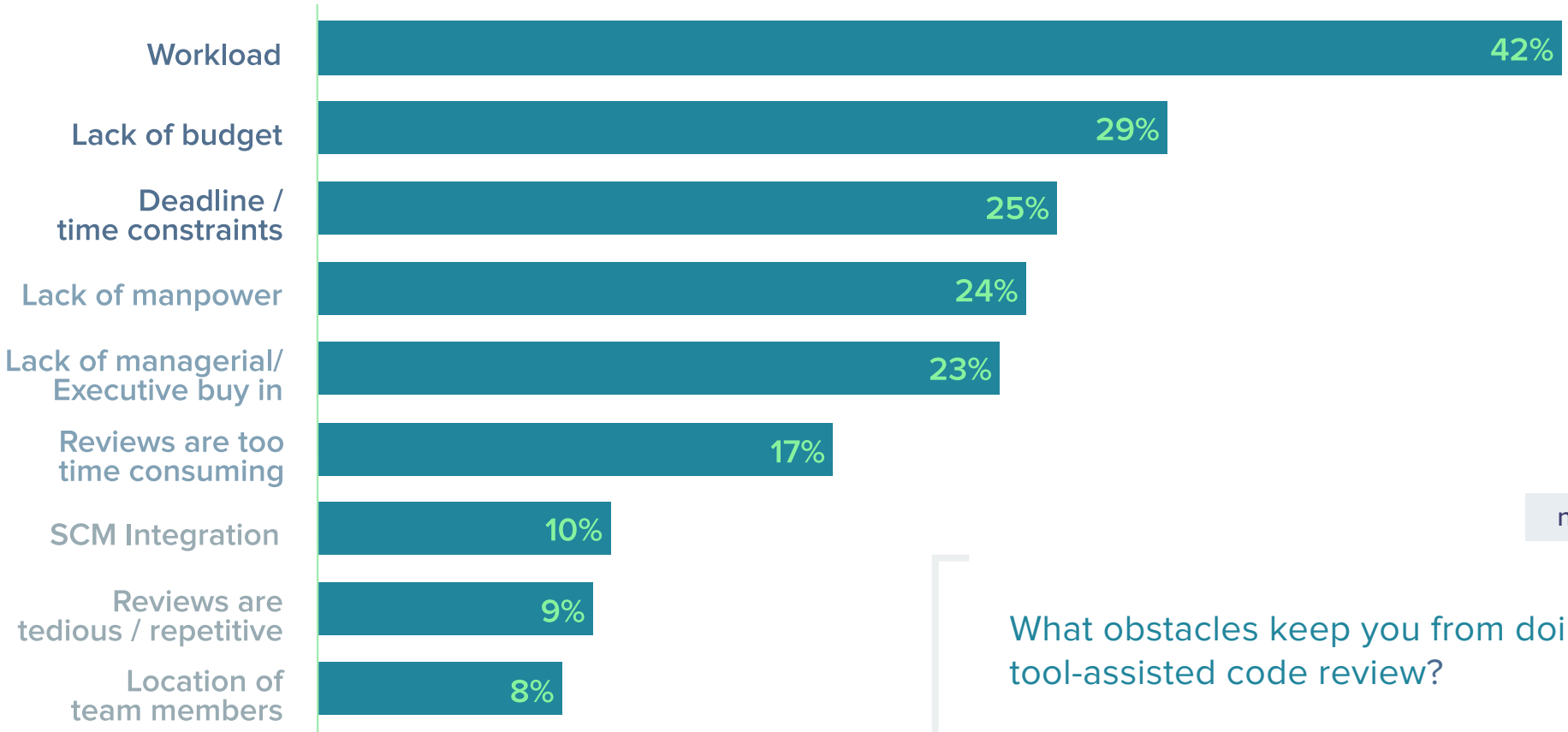
n = 493 *by Company Size*



Workload, budget, and time constraints are the biggest obstacles to doing tool-assisted code review

With nearly 40% of respondents not currently using a tool for code review, we wanted to better understand the factors that keep teams from adopting a code review tool. Similar to the obstacles team face with doing code review, workload, time constraints, and manpower are major obstacles holding teams back.

Budget and lack of managerial buy in were also viewed as blockers keeping teams from adopting code review tools.



n = 508

What obstacles keep you from doing tool-assisted code review?

Section 4: Software Tools & Integrations

These days code review is just one of the processes that go into developing and maintaining software quality. Software teams depend on a number of tools and systems for software development. In section 4 of the *State of Code Quality 2016* report, we look at trends within the software industry, related to the adoption of software tools and systems.

Highlights

01 Git is the most commonly used software configuration management tools (SCMs), with 43.4% of respondents now using Git.

02 61% of respondents are not using a repository management tool.

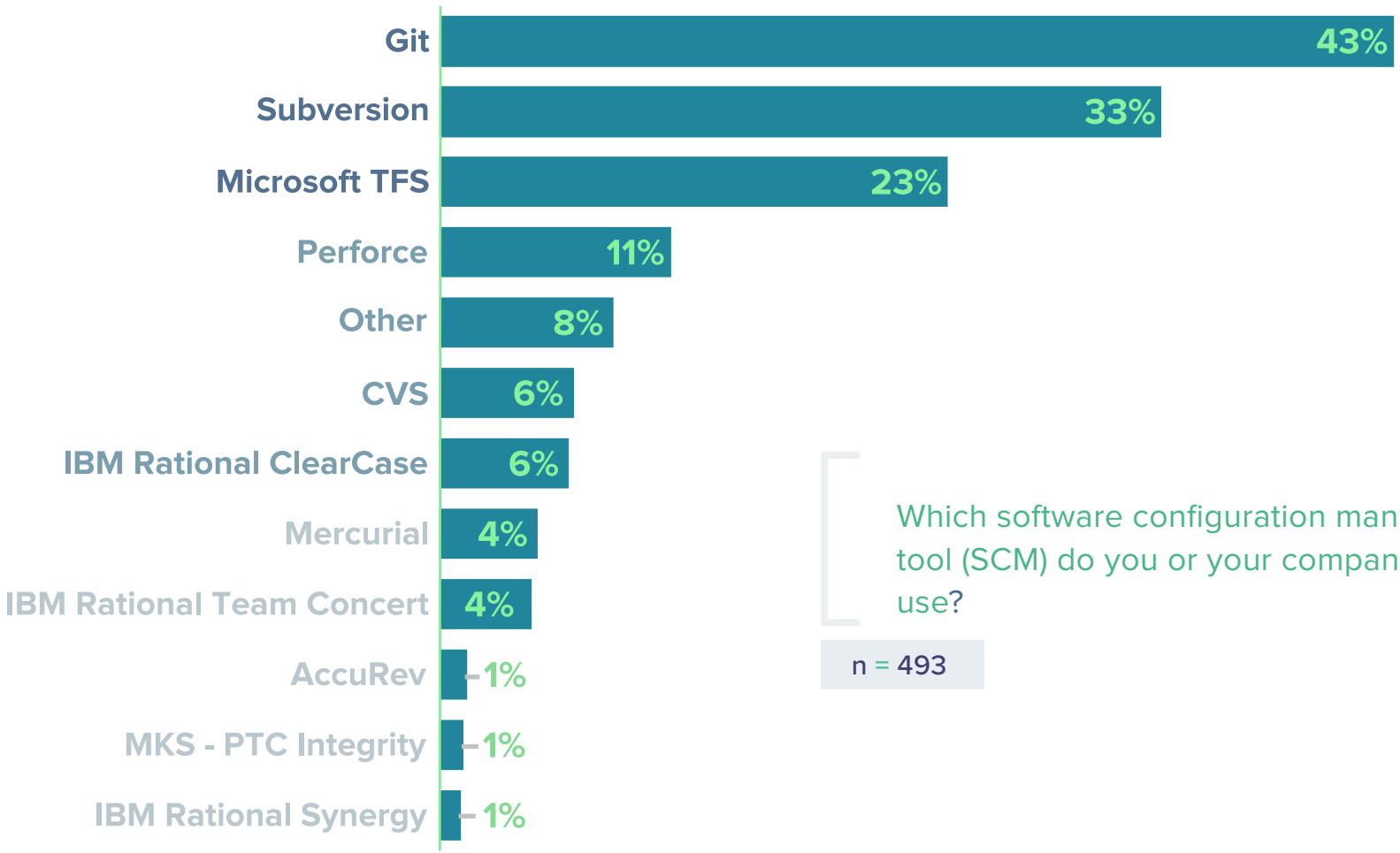
03 More than three-quarters of respondents are not planning on introducing a repository management tool in the next three months.

04 37% of respondents are using Visual Studio as their integrated development environment; 1 in 4 are using Eclipse.

05 JIRA is the most commonly used tool for bug tracking (41.8%). JIRA is also the top tool for requirements management (34%).

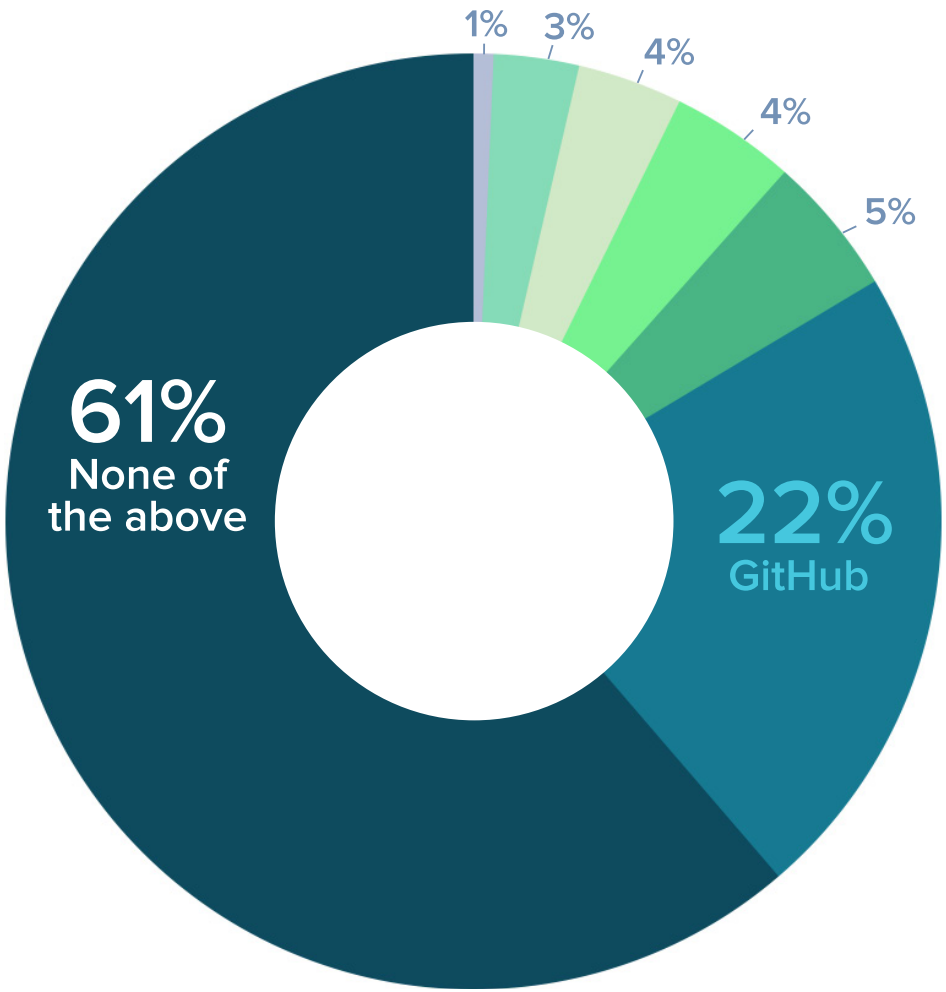
Git is the most commonly used software configuration management tool, with 43.4% of teams using it

One-third of development teams are using Subversion and nearly a quarter are using Microsoft TFS. 54% of companies with 100 employees or less are using Git.



Which software configuration management tool (SCM) do you or your company currently use?

n = 493



61% of respondents are not using a repository management tool; 1 in 5 are using GitHub

61% of respondents are not using any of the well-known repository management tools. Of those who are using a repository management tool, most are using GitHub, GitHub Enterprise, or GitLab.

Are you currently using any of the following repository management tools?

n = 579

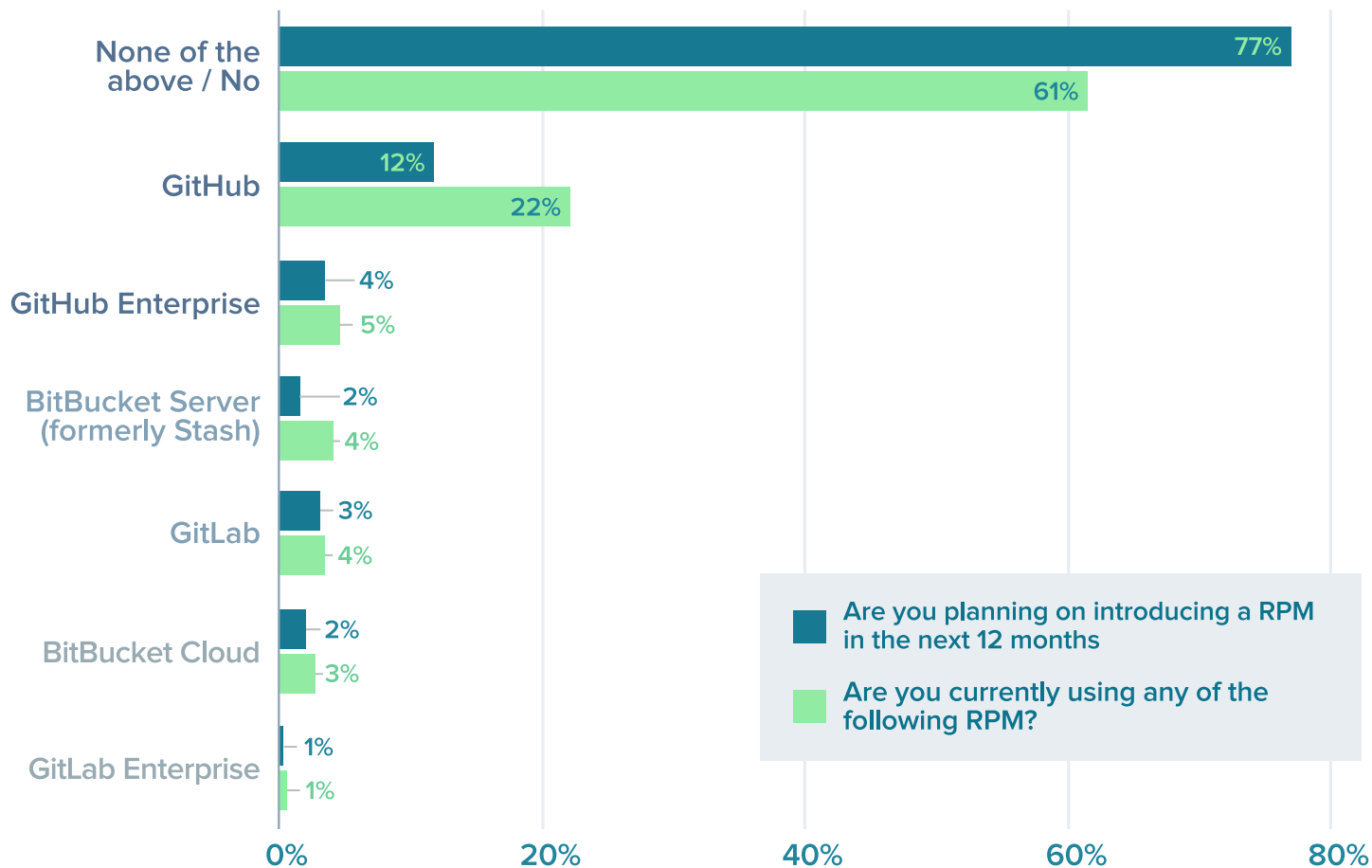
- None of the above
- GitHub
- GitHub Enterprise
- BitBucket Server (formerly Stash)
- GitLab
- BitBucket Cloud
- GitLab Enterprise

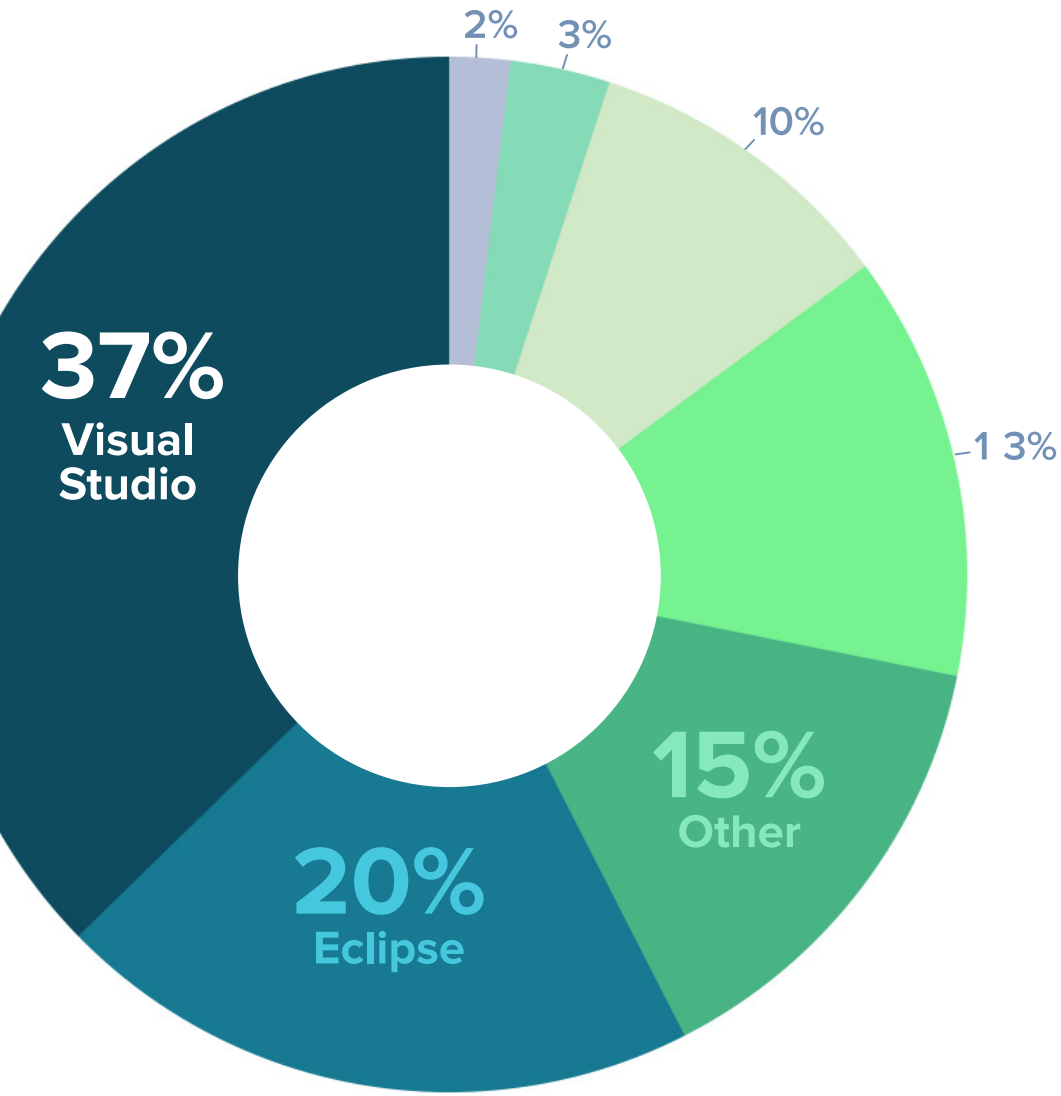
More than three-quarters of respondents are not planning on introducing a repository management system in the next three months

Of those respondents that are planning on introducing a repository management system, a majority are planning on implementing Github or Github Enterprise.

Are you planning on introducing any repository management tools in the next 12 months?

n = 579





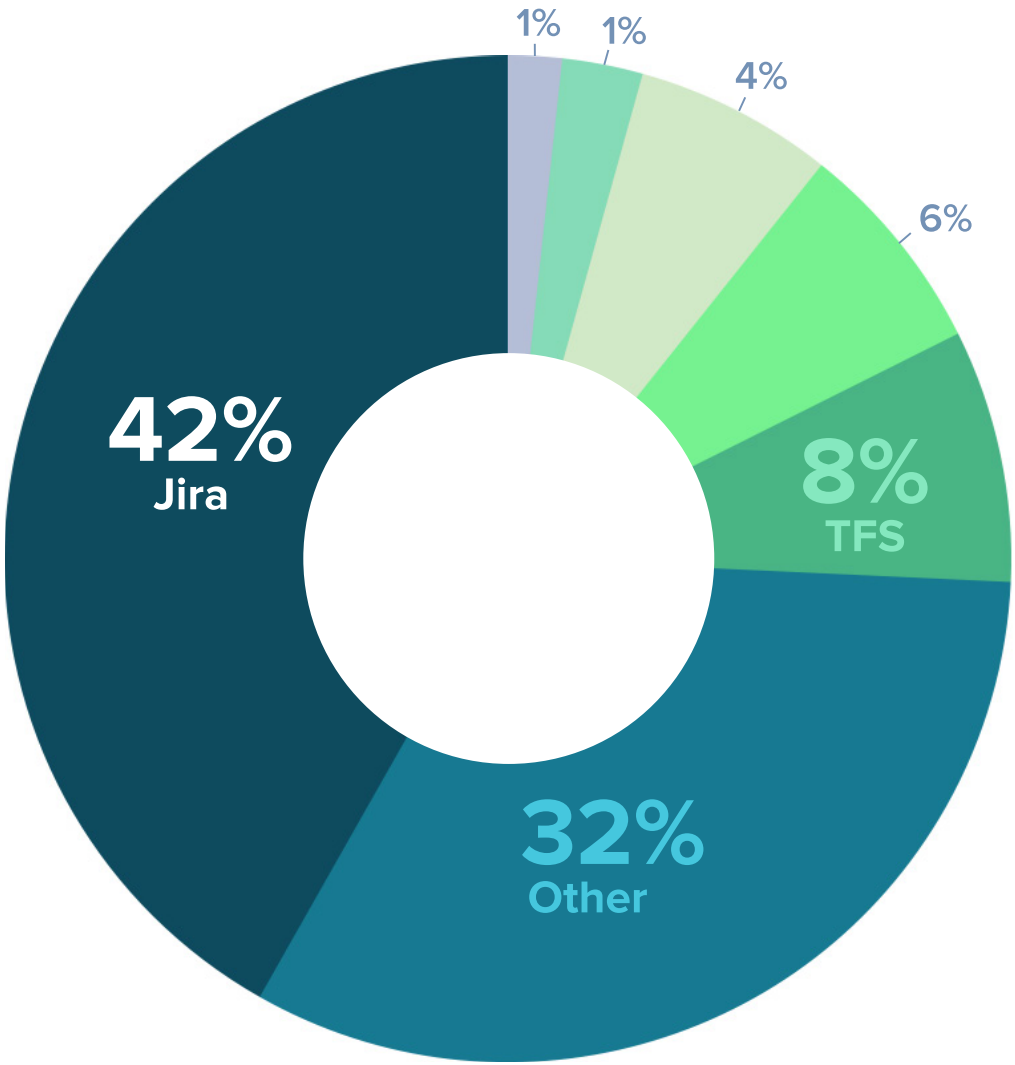
37% of respondents are using Visual Studio as their integrated development environment; 1 in 5 are using Eclipse

A vast majority of respondents are using an integrated development environment (87%) — with more than a third using Visual Studio and 1 in 5 using Eclipse.

Which integrated development environment are you using?

n = 579

- Visual Studio
- Eclipse
- Other
- Not Using One
- IntelliJ
- NetBeans
- XCode



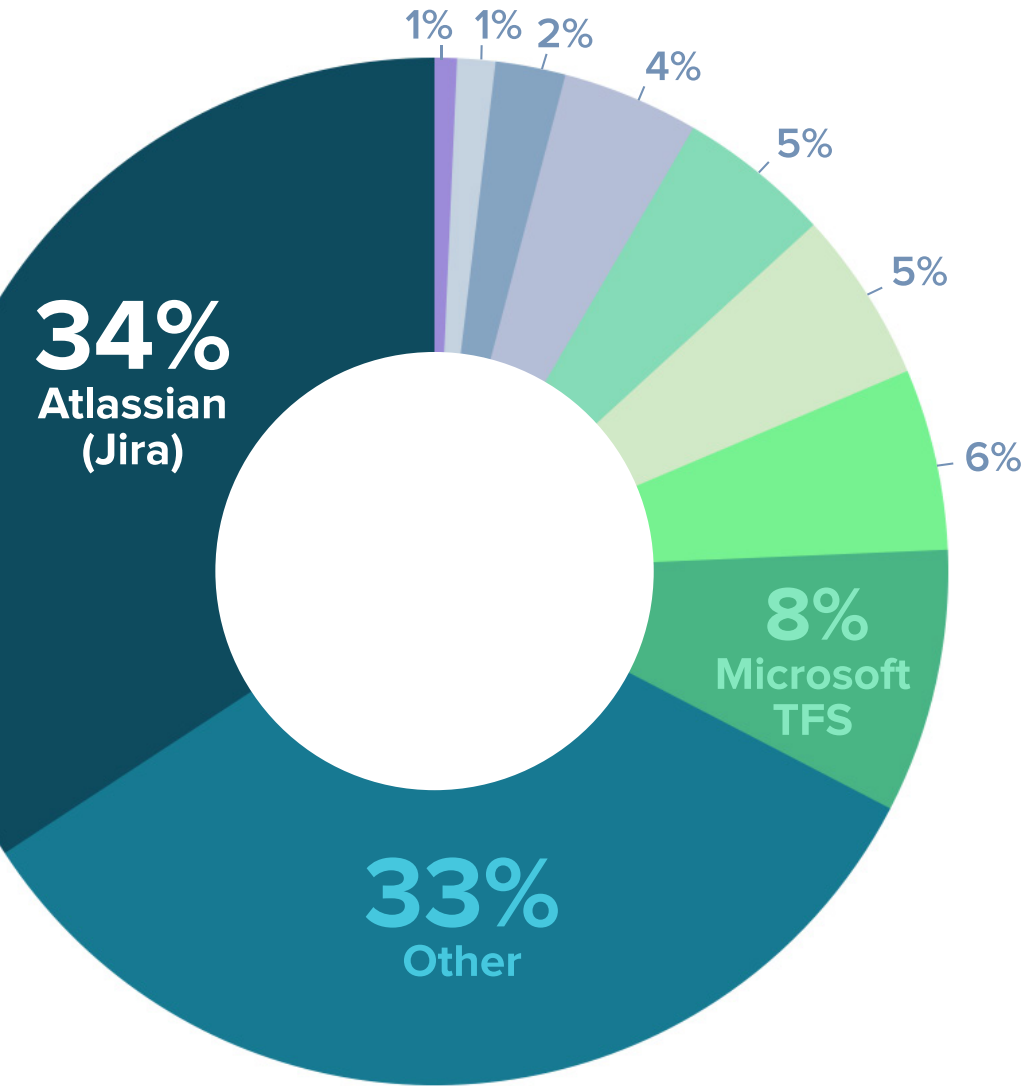
JIRA is the most commonly used tool for bug tracking, with 41.8% of respondents using JIRA

JIRA outpaced all other bug tracking tools by at least a 4:1 margin. For organizations that aren't using JIRA, there is a lot of variation in how they are tracking bugs, with no clear runner up to JIRA.

What tool are you using for bug tracking?

n = 579

- Jira
- Other
- TFS
- HP ALM
- BugZilla
- Mantis
- FogBugz



JIRA is the most commonly used tool for requirements management, with 34% of organizations using JIRA

Similar to what we saw with bug tracking tools, there is also a good amount of variation in the tools organizations are using for requirements management. While JIRA is being used by one-third of organizations, there were no other clear standout tools for requirements management.

7% of respondents reported that they are not currently using a tool for requirements management.

What tool are you using for requirements management?

n = 579

- Atlassian (Jira)
- Other
- Microsoft TFS
- HP ALM
- VersionOne
- Rally
- IBM Rational DOORS
- 4IBM Rational DOORS Next Gen
- TestTrack RM
- PTC Integrity

Section 5: Software Teams

The *State of Code Quality 2016* report includes responses from software developers, testers, IT/operations professionals, and business leaders representing more than 30 different industries. Participants in the survey work on software teams ranging from less than 5 employees up to more than 51 employees, and work for companies ranging from small businesses, with less than 25 employees, to enterprise organizations, with 10,000 employees or more.

Highlights

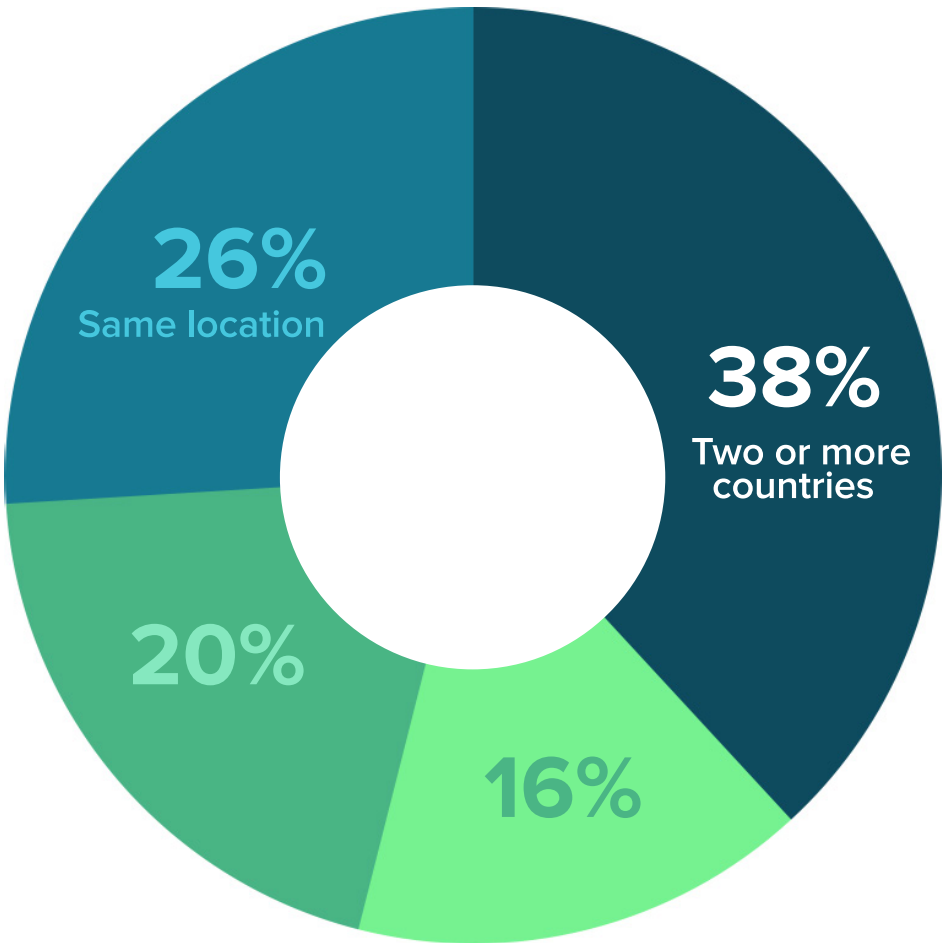
01 More than half of teams are distributed across multiple locations; 38% are international.

02 45% of respondents work in companies with more than 10,000 employees; 1 in 4 work in companies with less than 100.

03 78% of respondents work on development teams with 20 members or less.

04 More than 30 industries are represented in the *State of Code Quality 2016* report.

05 48% of survey respondents are developers; 17% are testers.



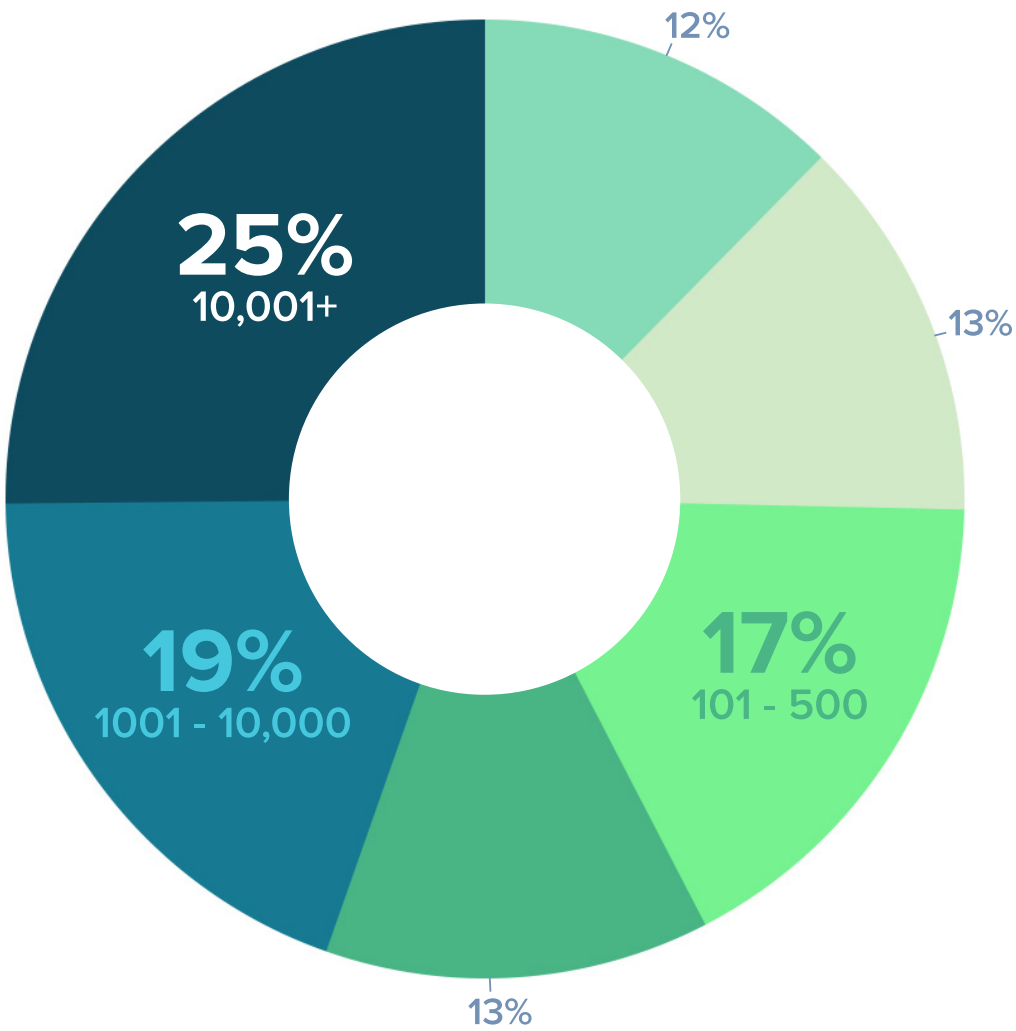
More than half of teams are distributed across multiple locations; 38% are international

Looking at how teams are distributed, we found that a majority of teams are now working across multiple locations. A majority of teams that are distributed across multiple locations are also working across two or more countries.

Which best describes your development team?

n = 493

- Two or more countries
- Same location
- Multiple sites, but the team is in one location
- Two or more locations in US



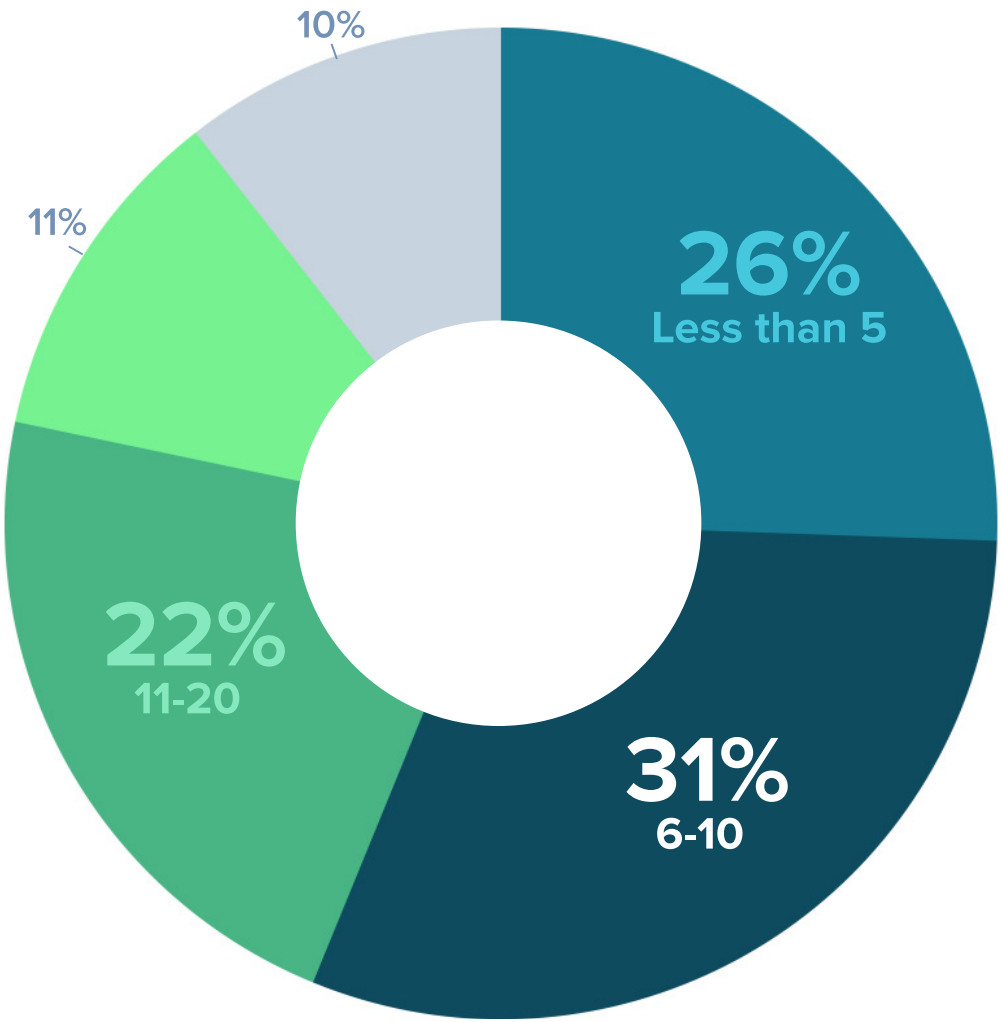
45% of respondents work in companies with more than 10,000 employees; 1 in 4 work in companies with less than 100

The *State of Code Quality 2016* report includes responses from software professionals working in small, medium, and enterprise organizations.

What is the total employee size of your company?

n = 493

- 10,001+
- 1001 - 10,000
- 501 - 1000
- 101 - 500
- 26 - 100
- Less than 25



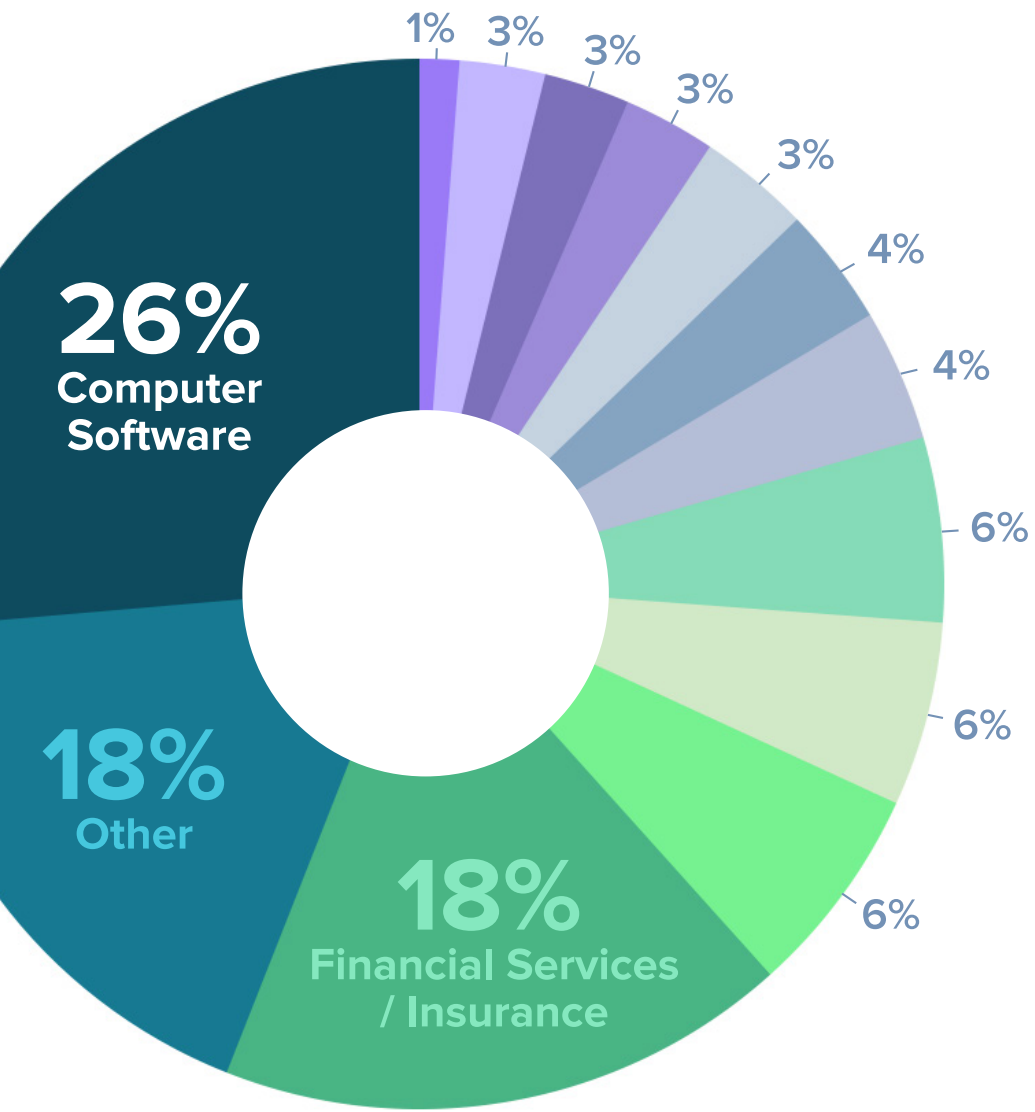
78% of respondents work on development teams with 20 members or less

Respondents in the *State of Code Quality 2016* survey work primarily on smaller development teams — with 1 in 4 are on teams with less than 5 employees.

What is the size of the development team you are on?

n = 493

- 51+
- 21 - 50
- 11 - 20
- 6 - 10
- Less than 5



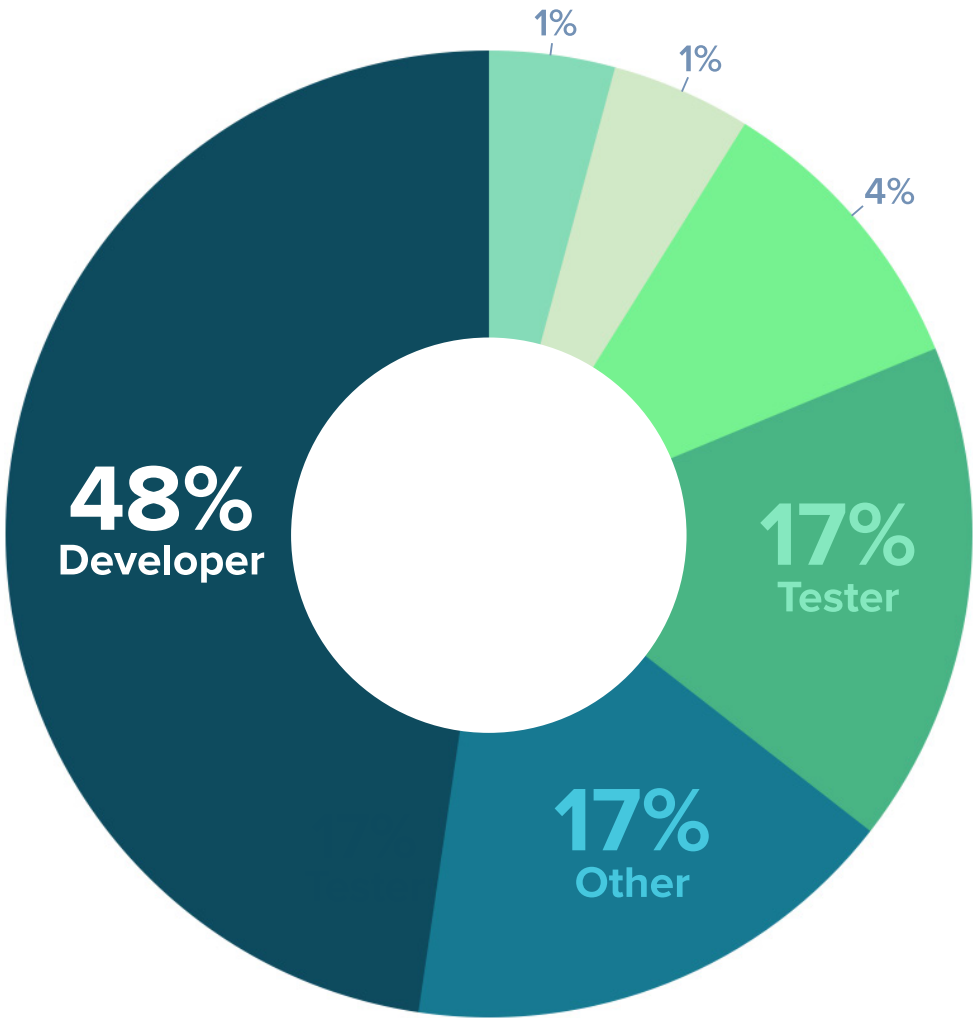
More than 30 industries are represented in the *State of Code Quality 2016* report

Computer software and financial services are the industries with the largest representation in the *State of Code Quality 2016* report.

What industry do you work in?

n = 493

- Computer Software
- Other
- Financial Services / Insurance
- Government / Federal
- Healthcare
- Retail / E-commerce
- Industrial Manufacturing
- Medical Devices
- Consulting / Systems Integrator
- Aerospace / Defense
- Mobile Development
- Game Development
- Marketing Services



48% of survey respondents are developers; 17% are testers

Development and QA were the roles best represented in the *State of Code Quality 2016* report.

What best describes your current role?

n = 619

- Developer
- Other
- Tester
- Development Manager
- Director / VP
- Testing Manager

SMARTBEAR

Over 4 million software professionals and 25,000 organizations across 194 countries use SmartBear tool

4M+ users

25K+ organizations

194 countries

See Some Succesful Customers >>

API READINESS

Functional testing through performance monitoring

SEE API READINESS PRODUCTS

TESTING

Functional testing, performance testing and test management

SEE TESTING PRODUCTS

PERFORMANCE MONITORING

Synthetic monitoring for API, web, mobile, SaaS, and Infrastructure

SEE MONITORING PRODUCTS

CODE COLLABORATION

Peer code and documentation review

SEE COLLABORATION PRODUCTS


Collaborator
SMARTBEAR