The Best Kept Secrets of Code Review

SMARTBEAR

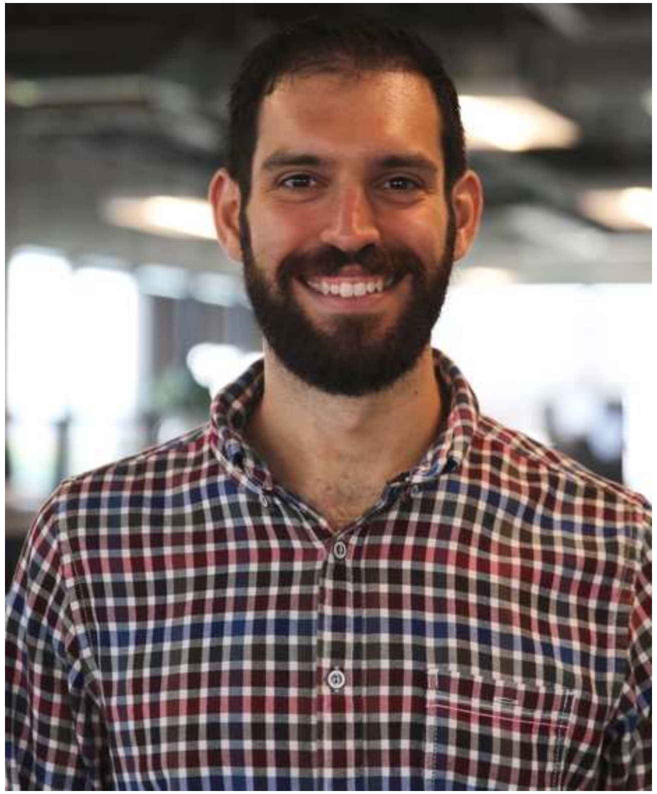
Your Hosts Today

Justin Collier

Product Owner, Collaborator

Patrick Londa

Marketing Manager, Collaborator



Jonathan Fortunati

Sr. Sales Engineer, Collaborator

2009

founded

11

global offices

6.5M+

users

22,000+

customers

5

free tools

525+

employees

SMARTBEAR

SmartBear Product Portfolio = Breadth **and** Depth

Design

- HipTest
- SwaggerHub
- Zephyr

Create

- ServiceV Pro
- Collaborator
- TestLeft

Test

- SoapUI Pro
- LoadUI Pro
- TestComplete
- LoadNinja
- CBT

Monitor

- AlertSite

Agenda

1 Why Focus on Code Reviews?

2 Code Review Approaches

3 Review Process Recommendations

4 Inside the Code Review

5 The Current Tool Landscape

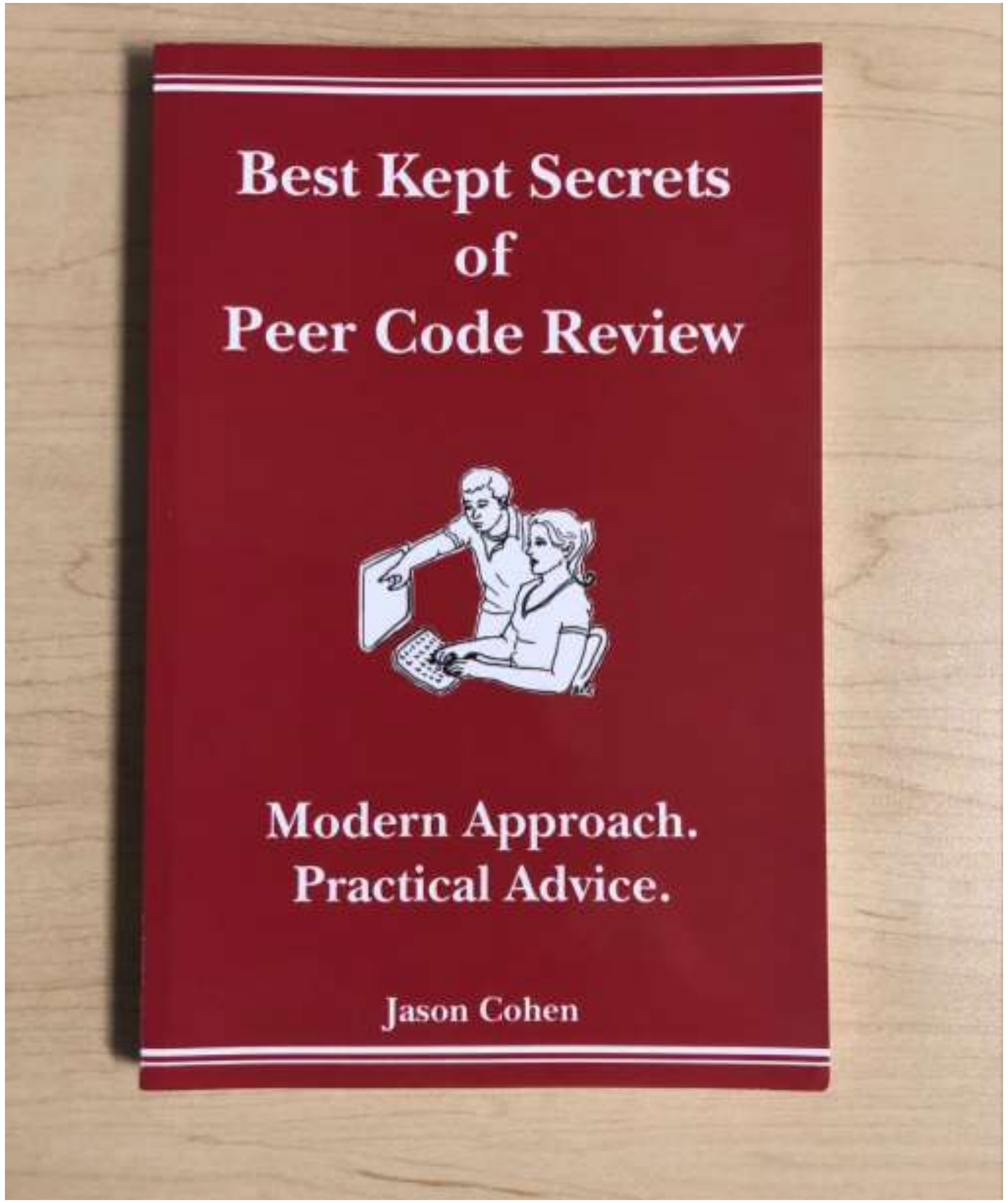
Main Sources

The State of Code Review

Trends and Insights into Collaborative Software Development

2018

SMARTBEAR



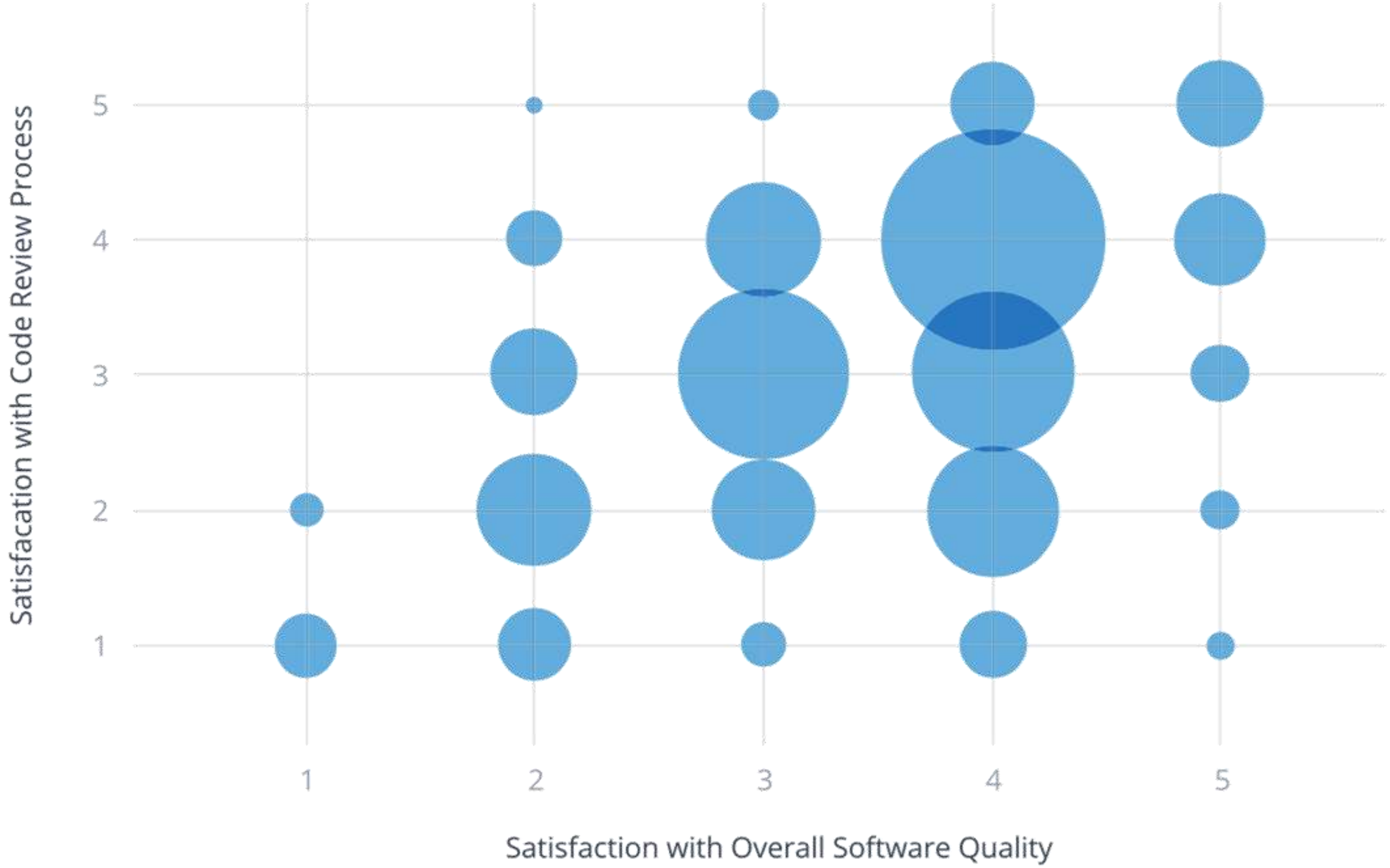
Best Kept Secrets of Peer Code Review

Modern Approach. Practical Advice.

Jason Cohen

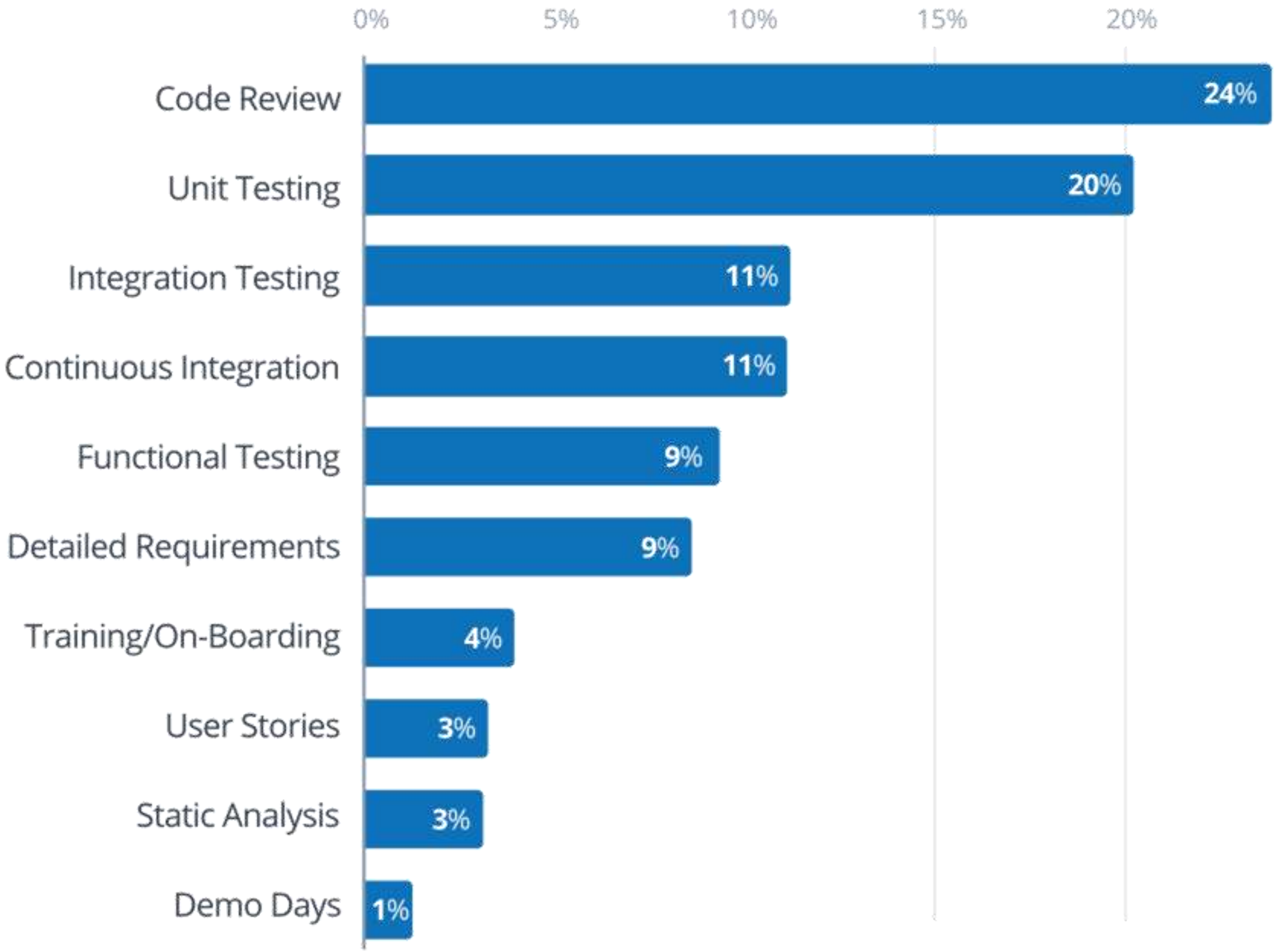
Why Focus on Code Review?

Correlation between code review process satisfaction and code quality satisfaction



The 2018 State of Code Review

What is the number #1 thing a company can do to improve code quality?



The 2018 State of Code Review

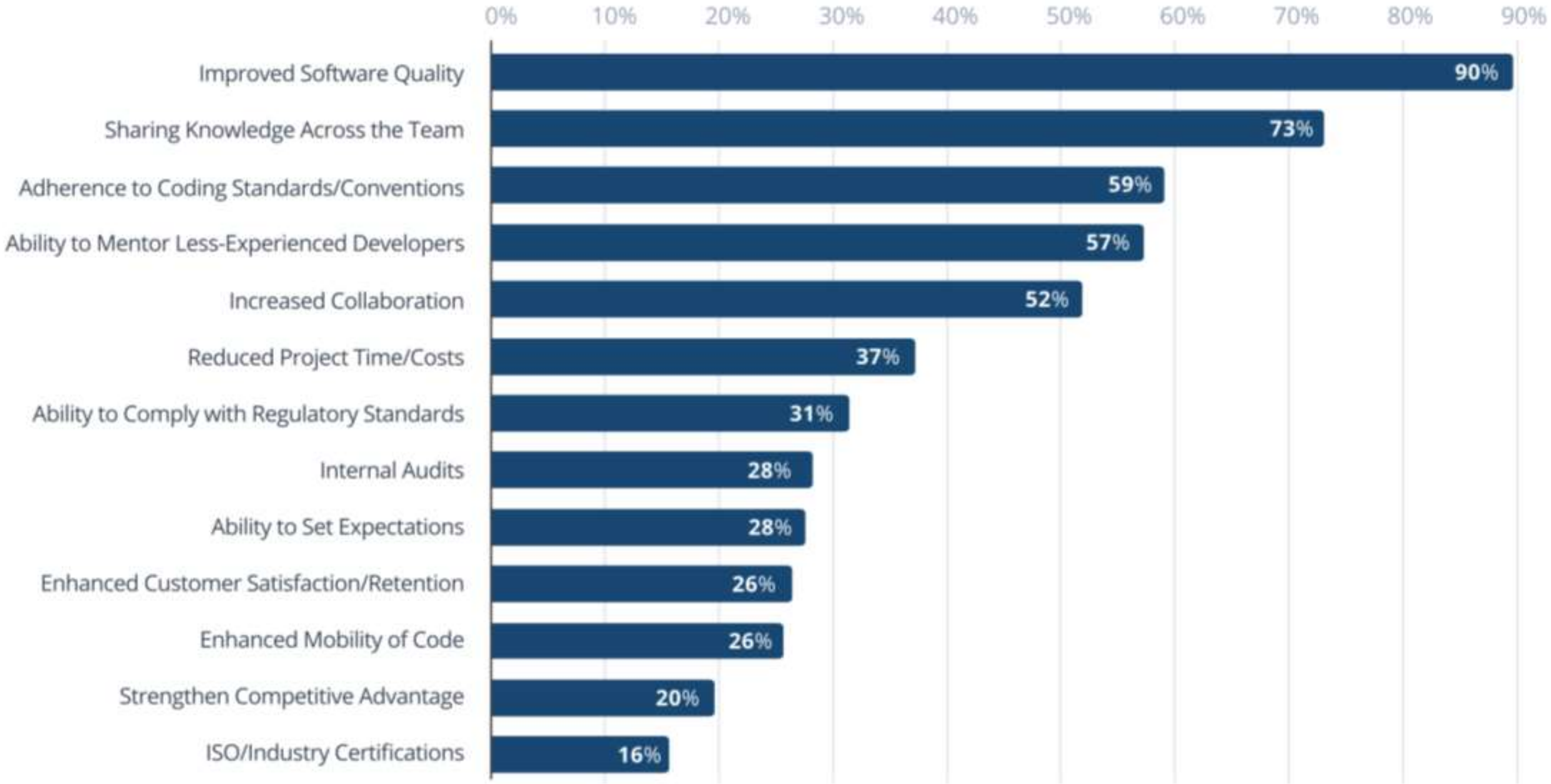
Code Review is the #1 Way to Improve Code Quality

According to: The 2018 State of Code Review

The 2017 State of Code Review

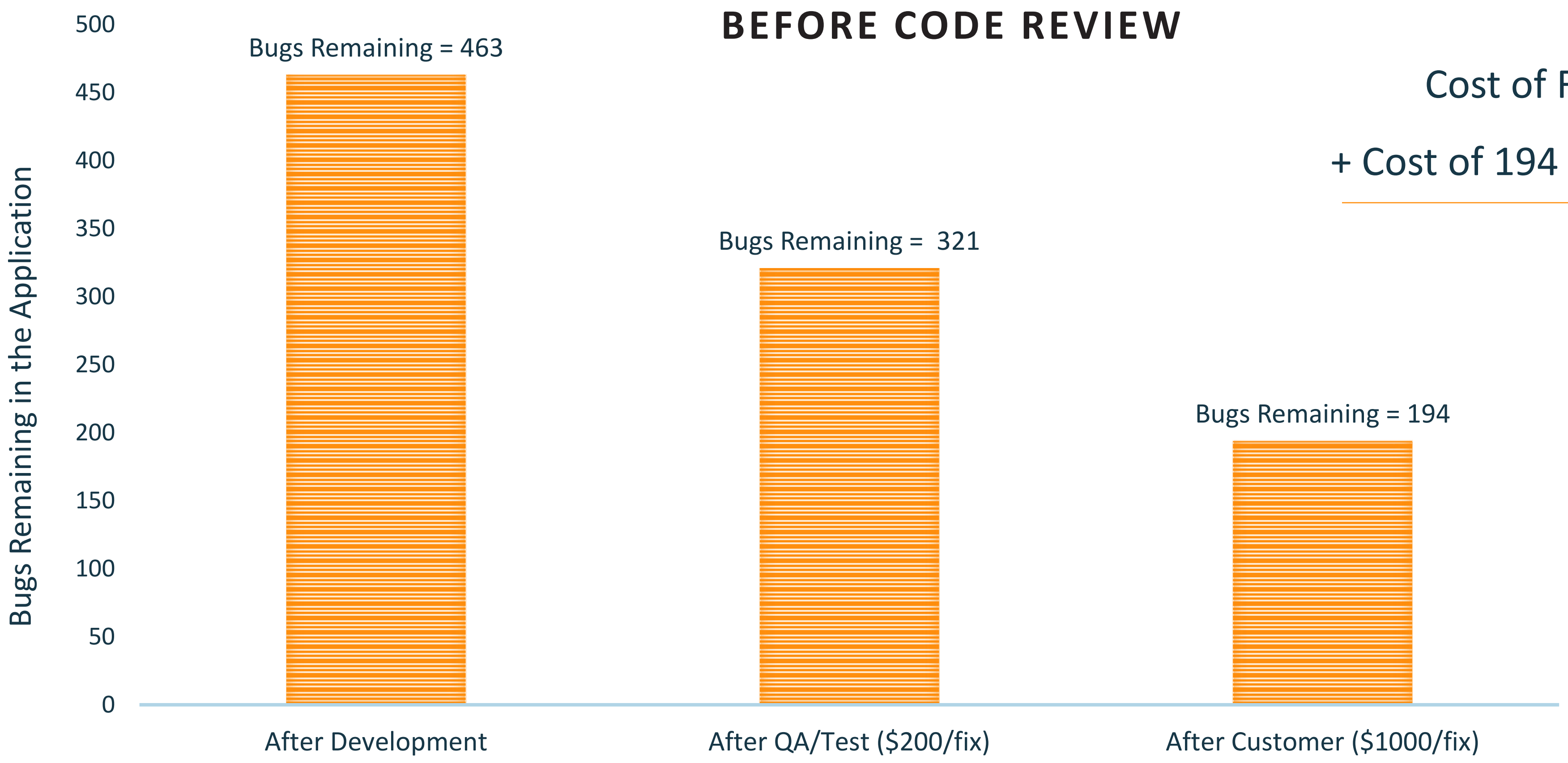
The 2016 State of Code Review

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



The 2018 State of Code Review

10,000-line project with 10 developers

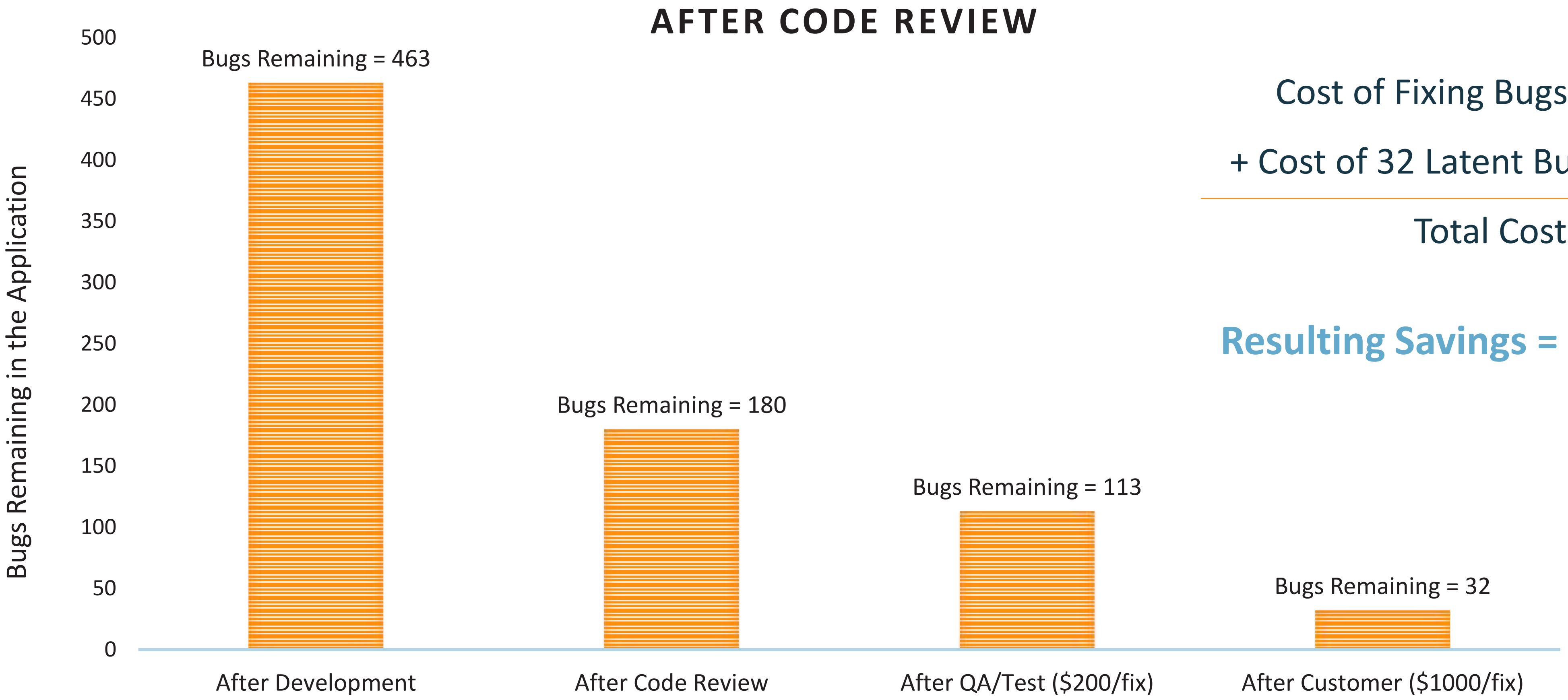


Cost of Fixing Bugs: $174k

+ Cost of 194 Latent Bugs: 194k

Total Cost: $**368k**

10,000-line project with 10 developers



Cost of Fixing Bugs: $120k

+ Cost of 32 Latent Bugs: 32k

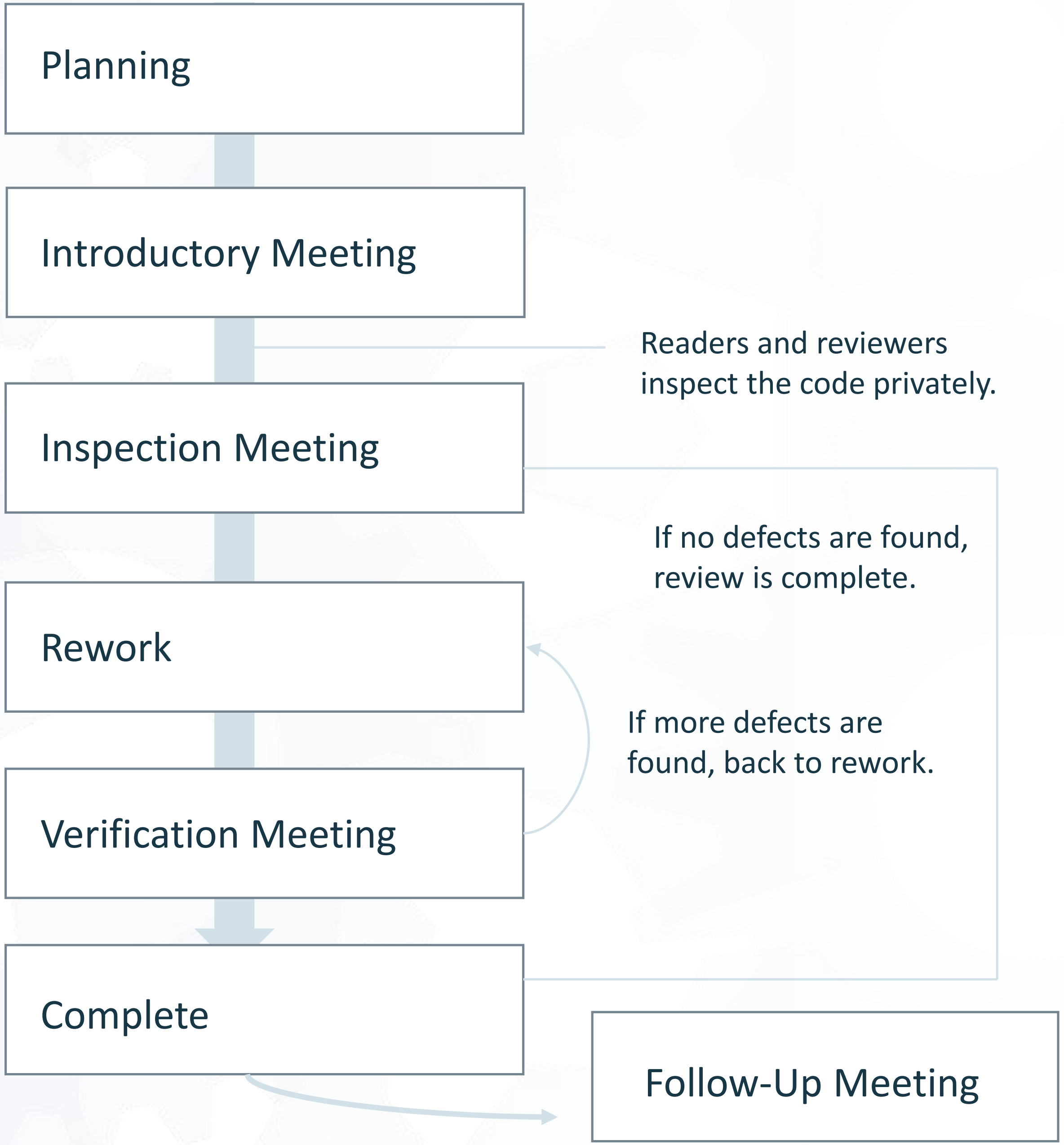
Total Cost: $**152k**

Resulting Savings = $216k

Code Review Approaches

Types of Code Review

- Formal Inspections



Planning

Introductory Meeting

Inspection Meeting

Rework

Verification Meeting

Complete

Follow-Up Meeting

Readers and reviewers inspect the code privately.

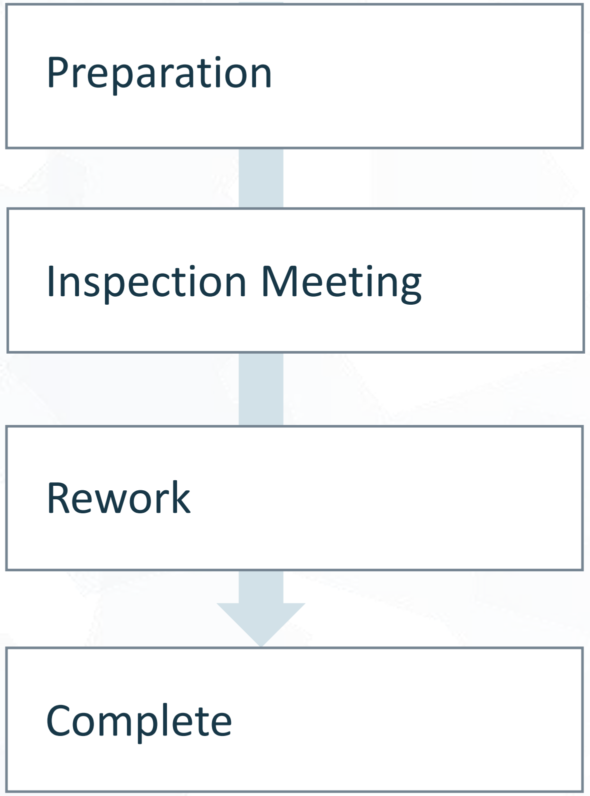
If no defects are found, review is complete.

If more defects are found, back to rework.

SMARTBEAR

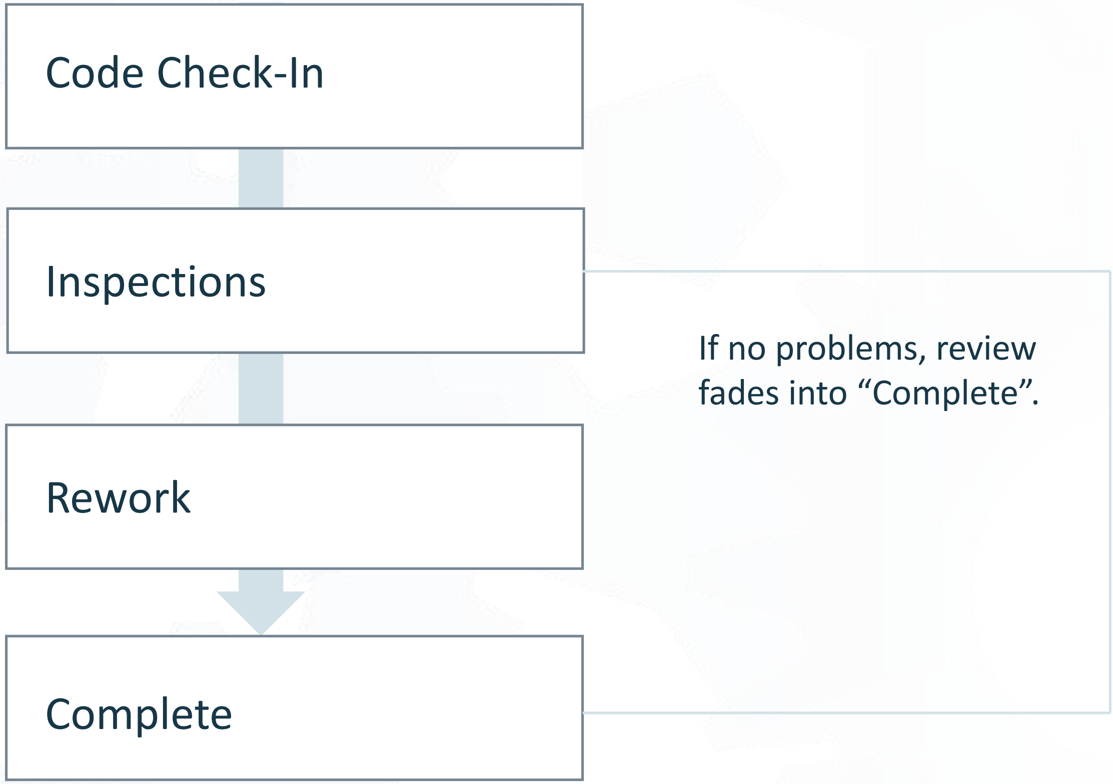
Types of Code Review

- Formal Inspections
- Over-the-Shoulder Reviews



Types of Code Review

- Formal Inspections
- Over-the-Shoulder Reviews
- Email Pass-Around Process



Code Check-In

Inspections

Rework

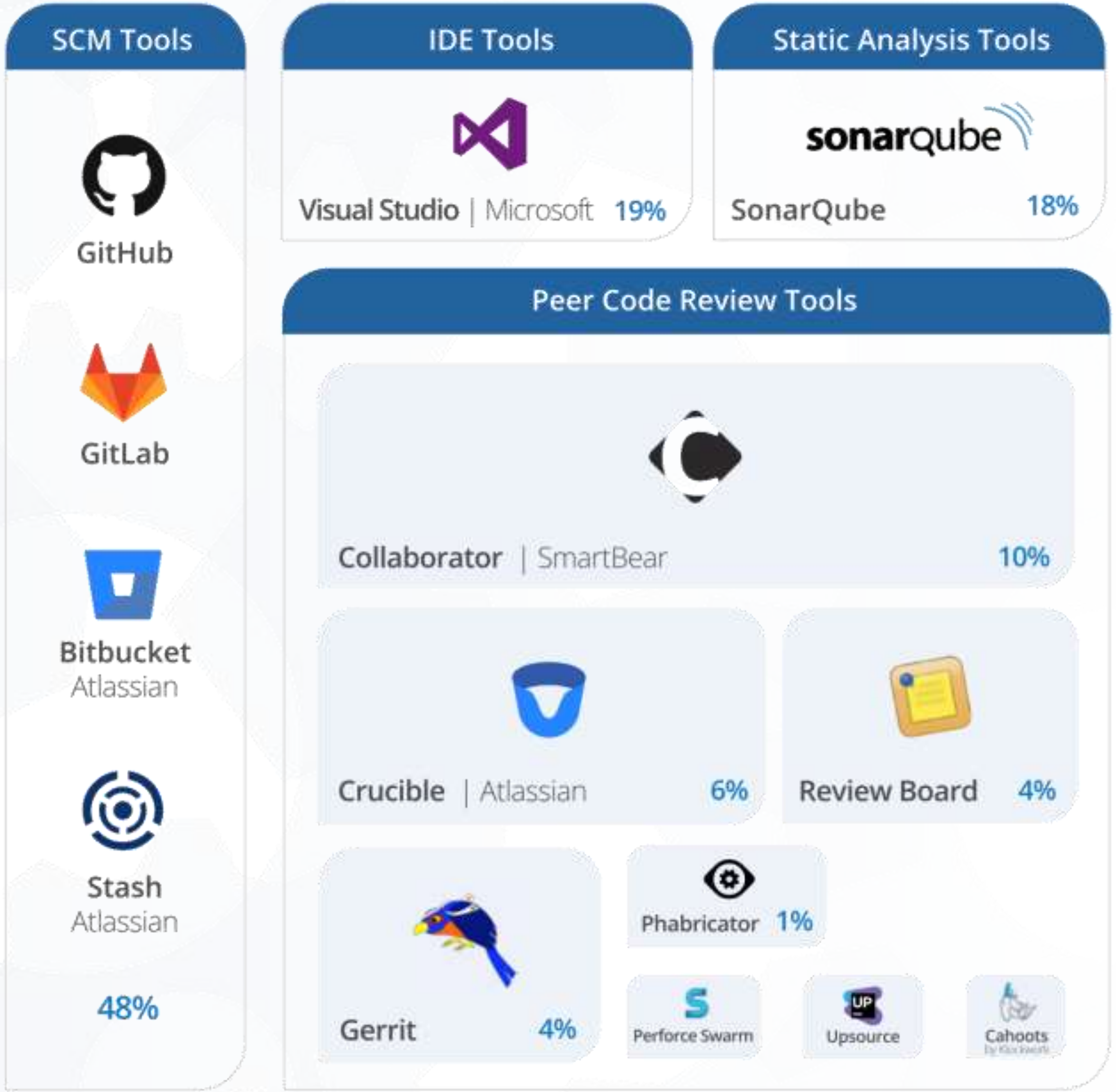
Complete

If no problems, review fades into “Complete”.

SMARTBEAR

Types of Code Review

- Formal Inspections
- Over-the-Shoulder Reviews
- Email Pass-Around Process
- Tool-Assisted Reviews



SCM Tools

GitHub

GitLab

Bitbucket
Atlassian

Stash
Atlassian

48%

IDE Tools

Visual Studio | Microsoft 19%

Static Analysis Tools

SonarQube 18%

Peer Code Review Tools

Collaborator | SmartBear 10%

Crucible | Atlassian 6%

Review Board 4%

Gerrit 4%

Phabricator 1%

Perforce Swarm

Upsource

Cahoots
by Klocwork

SMARTBEAR

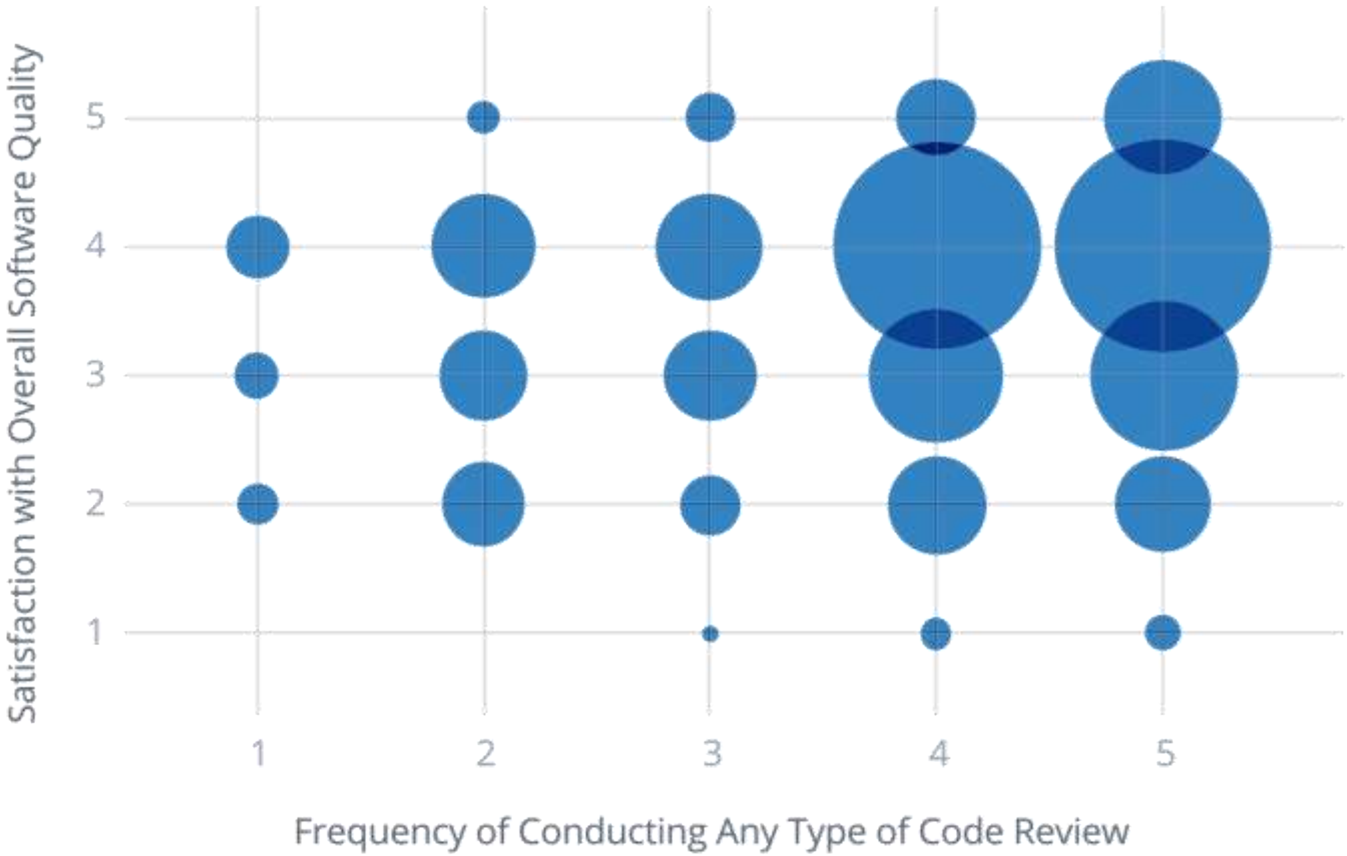
Types of Code Review

- Formal Inspections
- Over-the-Shoulder Reviews
- Email Pass-Around Process
- Tool-Assisted Reviews
- Pair Programming

Review Process Recommendations

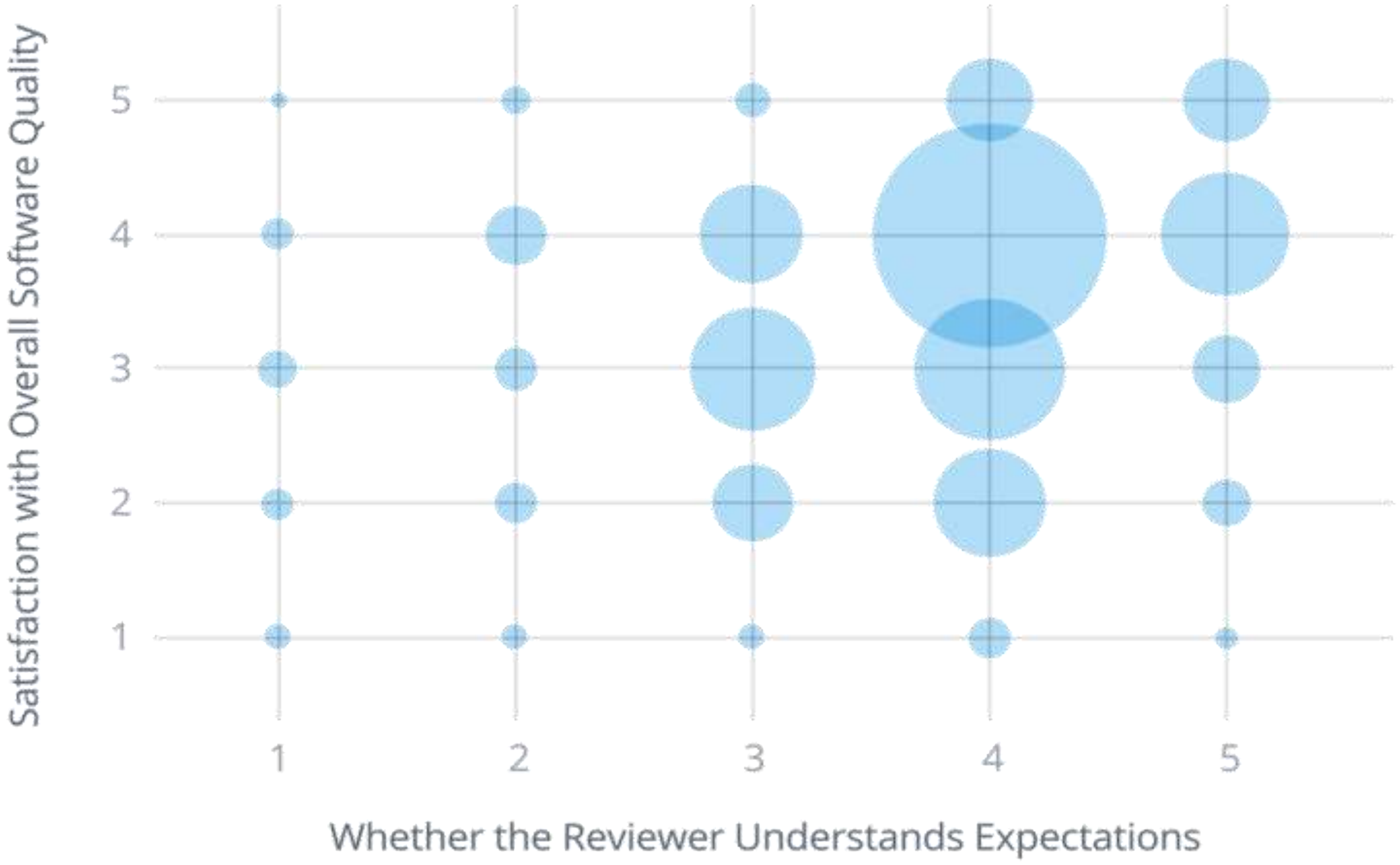
What Works Best:

- Conduct Reviews Daily



What Works Best:

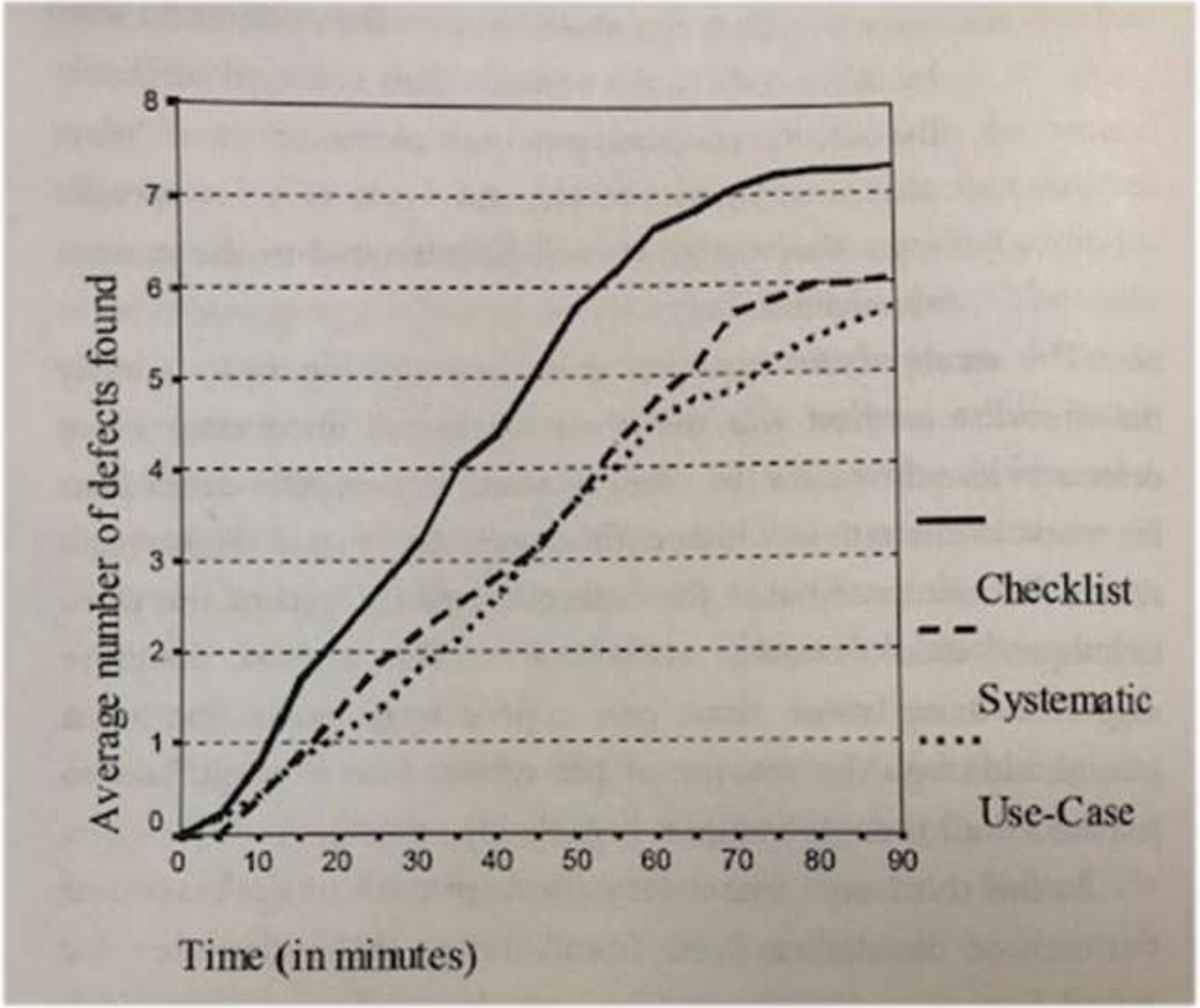
- Conduct Reviews Daily
- Set Clear Expectations



What Works Best:

- Conduct Reviews Daily
- Set Clear Expectations

***Utilize a checklist of things to check for at the class, method, & class-hierarchy levels**



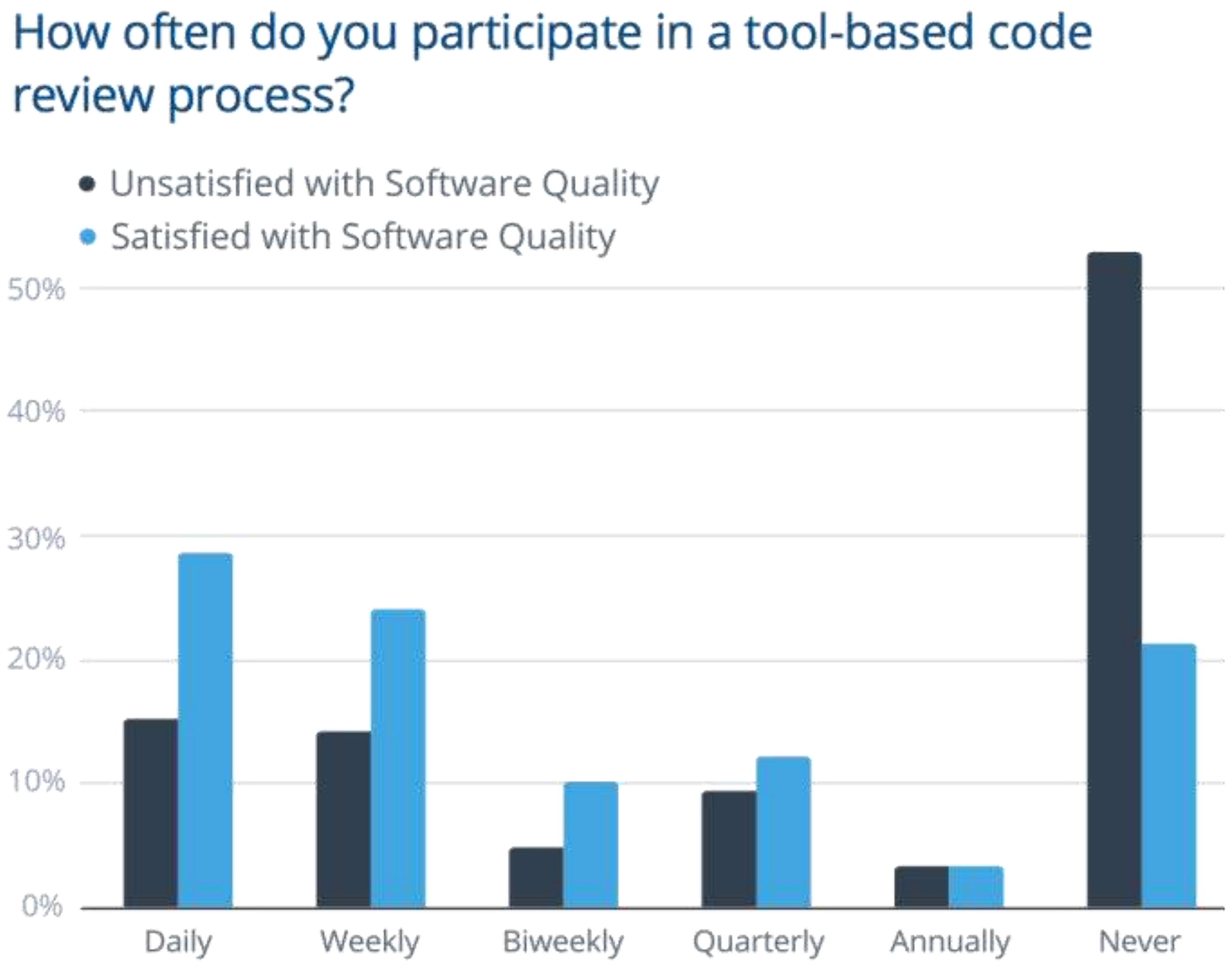
Research by Alastair Dunsmoore, Marc Roper, and Murray Wood. (2000)

The Best Kept Secrets of Peer Code Review

SMARTBEAR

What Works Best:

- Conduct Reviews Daily
- Set Clear Expectations
- Go Tool-Assisted



SMARTBEAR

The 2018 State of Code Review

What Works Best:

- Conduct Reviews Daily
- Set Clear Expectations
- Go Tool-Assisted + Meetings

We never perform meeting-based reviews...

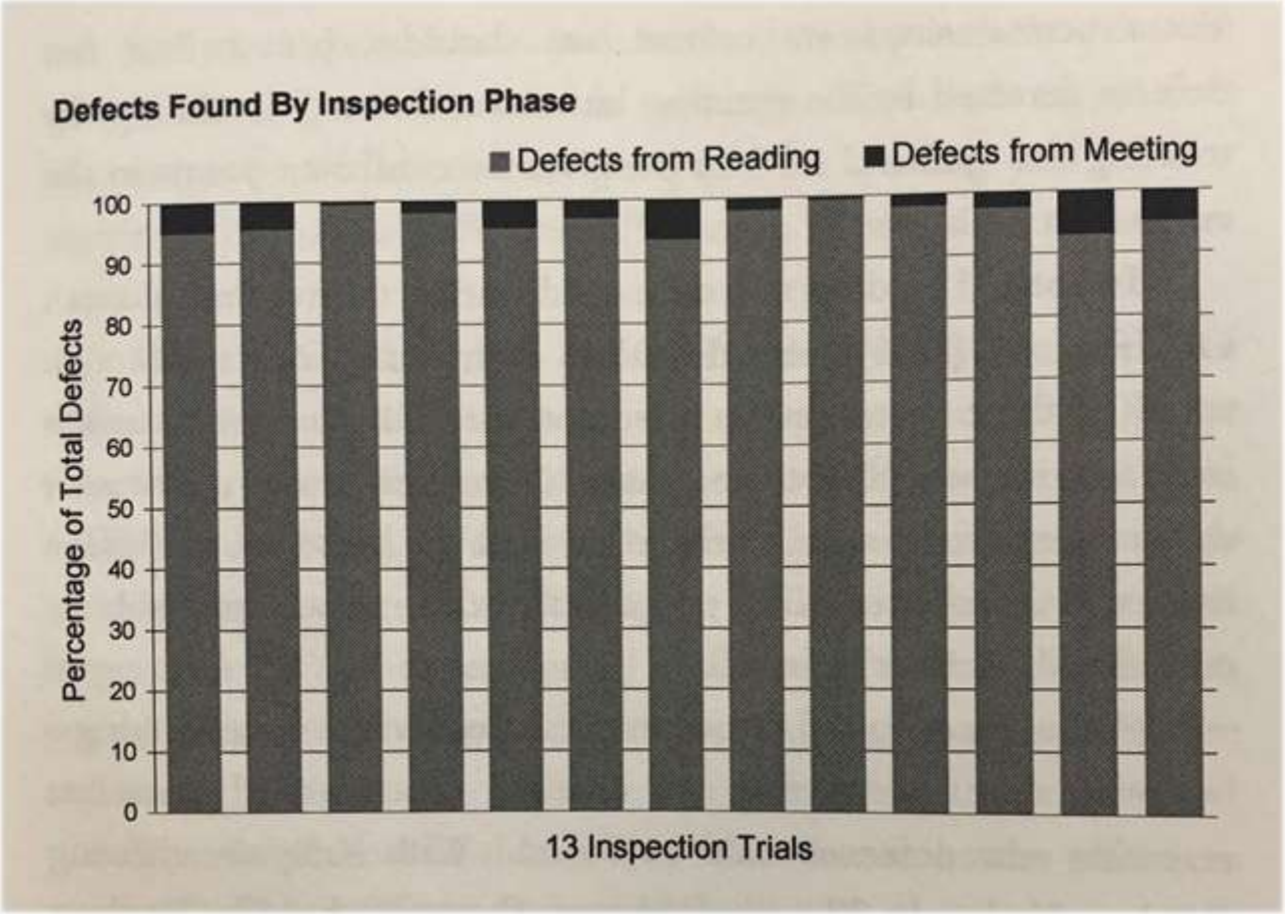
28% Satisfied with Code Quality

48% Unsatisfied with Code Quality

The 2018 State of Code Review

What Works Best:

- Conduct Reviews Daily
- Set Clear Expectations
- Go Tool-Assisted + Meetings



Research by Lawrence Votta from AT&T Bell Laboratories. (1993)
Supporting research by Diane Kelly and Terry Shepard. (2003)
Supporting research by Reidar Conradi. (2003)

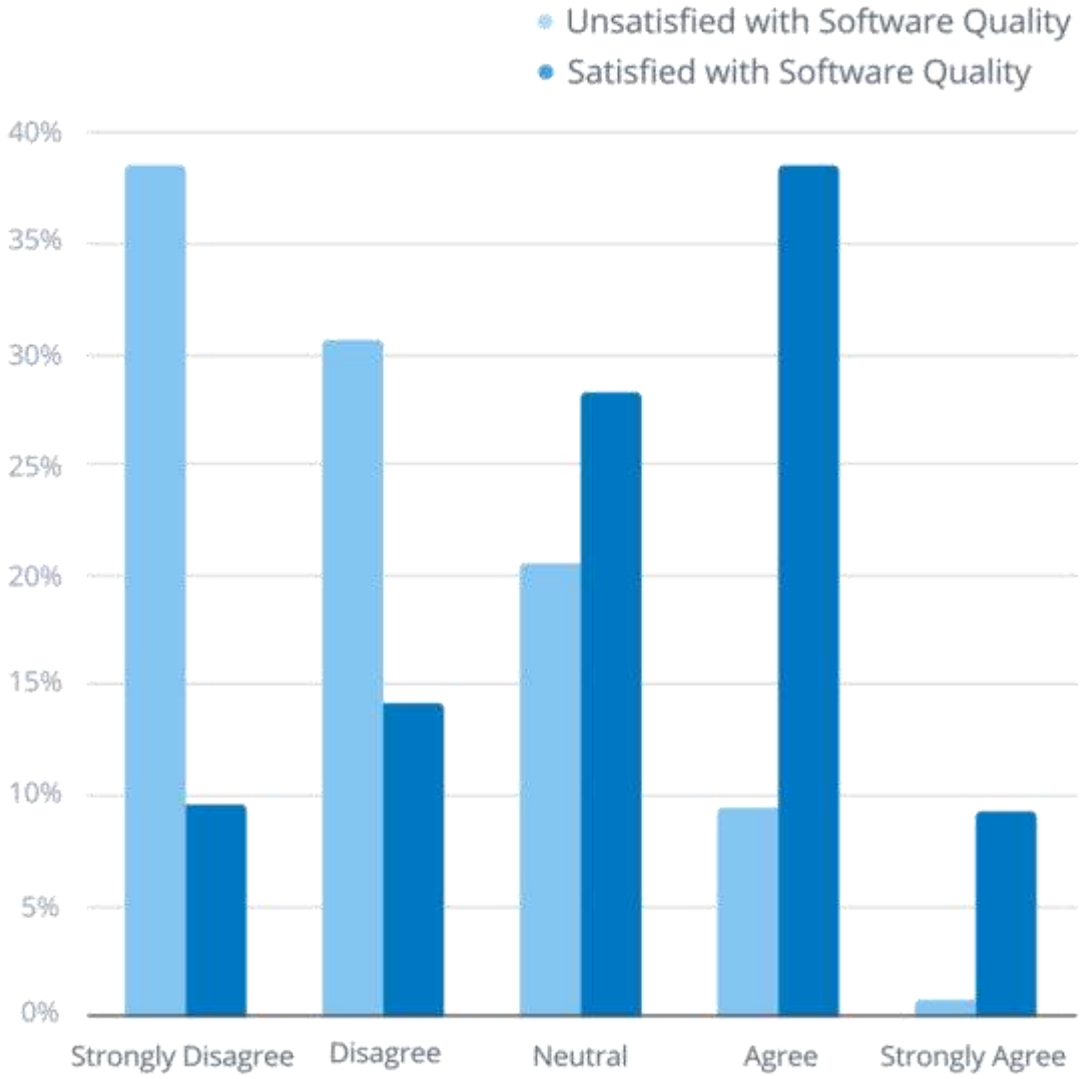
SMARTBEAR

The Best Kept Secrets of Peer Code Review

What Works Best:

- Conduct Reviews Daily
- Set Clear Expectations
- Go Tool-Assisted + Meetings
- Report on Their Process

My team regularly pulls reports and metrics on our code review process.



SMARTBEAR

The 2018 State of Code Review

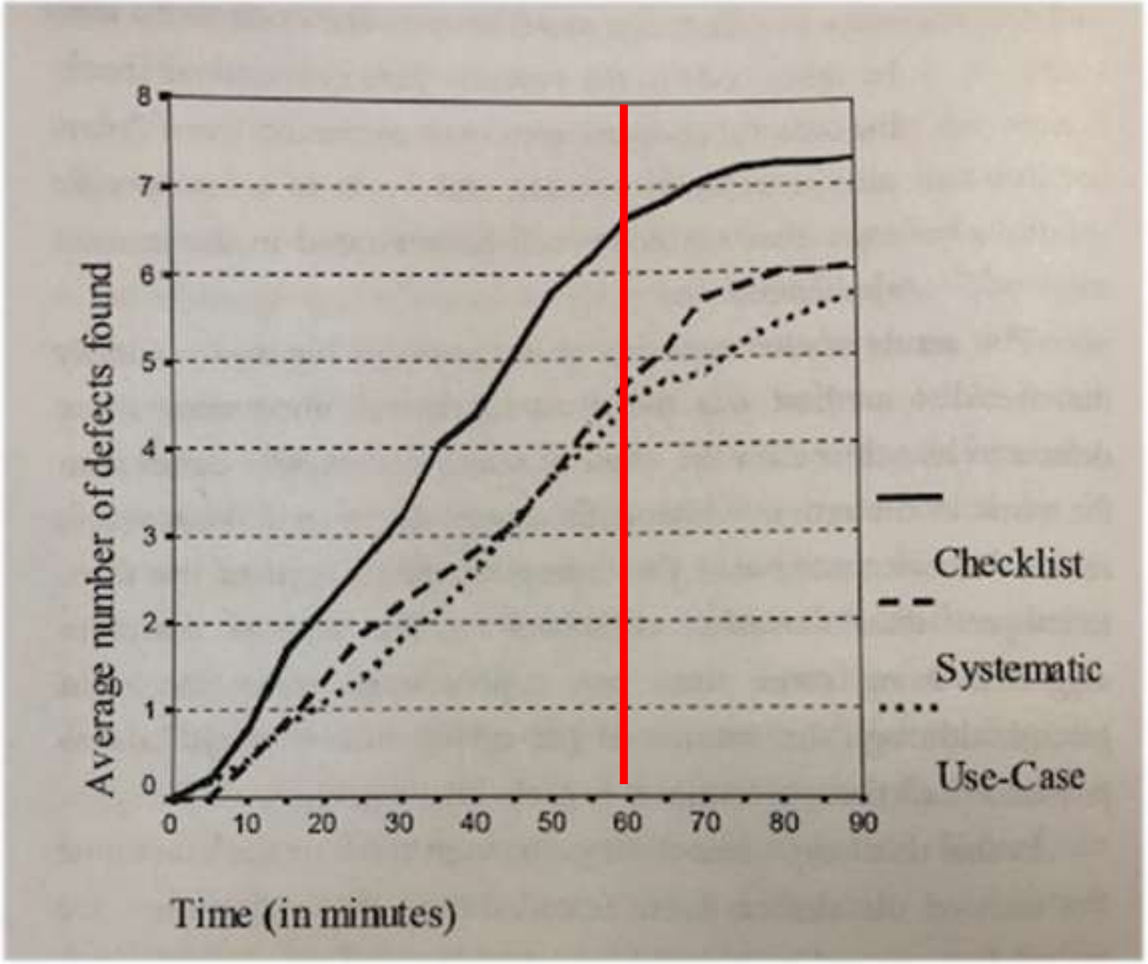
What Works Best:

- Conduct Reviews Daily
- Set Clear Expectations
- Go Tool-Assisted + Meetings
- Report on Their Process
- Foster a Learning Culture

Inside the Code Review

Review Guidelines

- Limit reviews to 60 minute sessions at most



Research by Alastair Dunsmoore, Marc Roper, and Murray Wood. (2000)

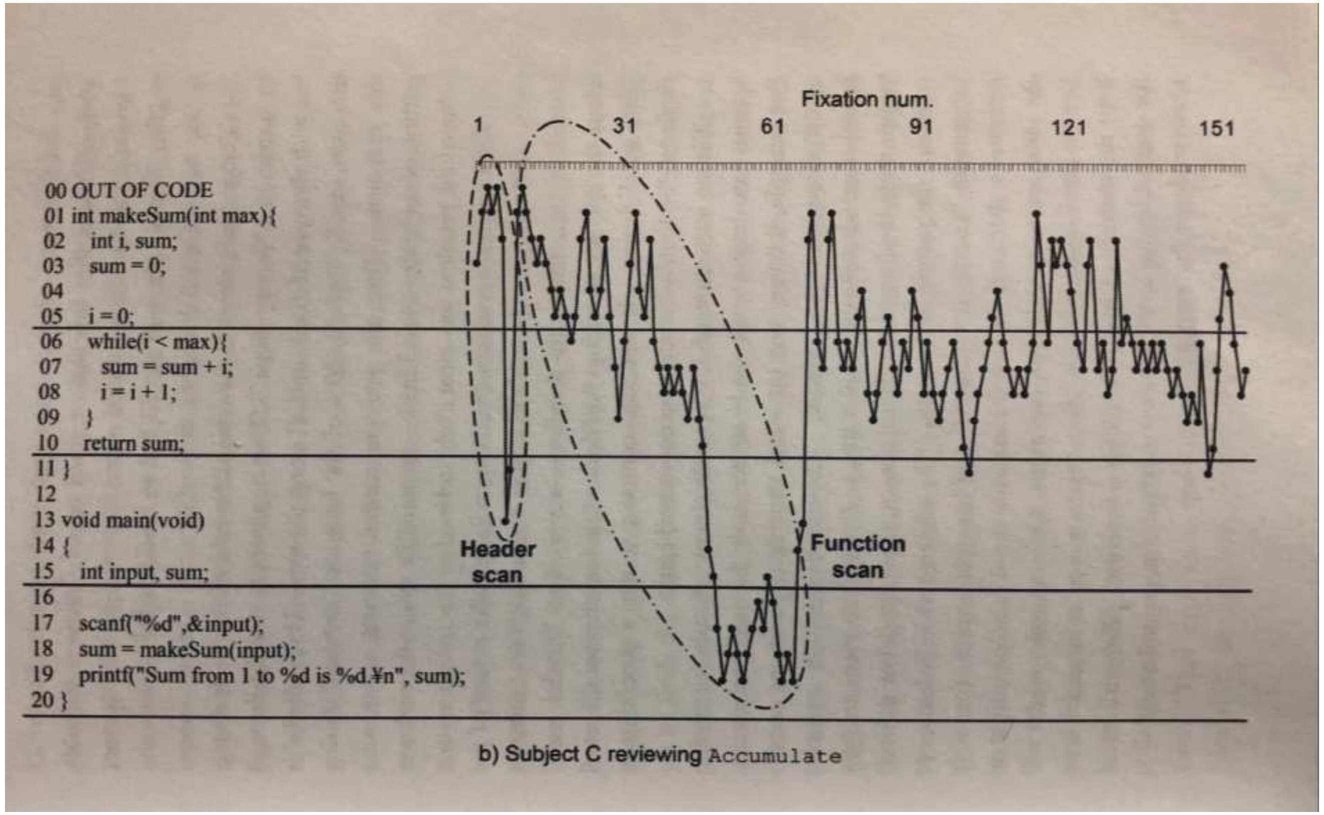
The Best Kept Secrets of Peer Code Review

SMARTBEAR

Review Guidelines

- Limit reviews to 60 minute sessions at most

- Consider shorter function length to improve readability



```
00 OUT OF CODE
01 int makeSum(int max){
02    int i, sum;
03    sum = 0;
04
05    i = 0;
06    while(i < max){
07       sum = sum + i;
08       i = i + 1;
09    }
10    return sum;
11 }
12
13 void main(void)
14 {
15    int input, sum;
16
17    scanf("%d",&input);
18    sum = makeSum(input);
19    printf("Sum from 1 to %d is %d.¥n", sum);
20 }
```

b) Subject C reviewing `Accumulate`

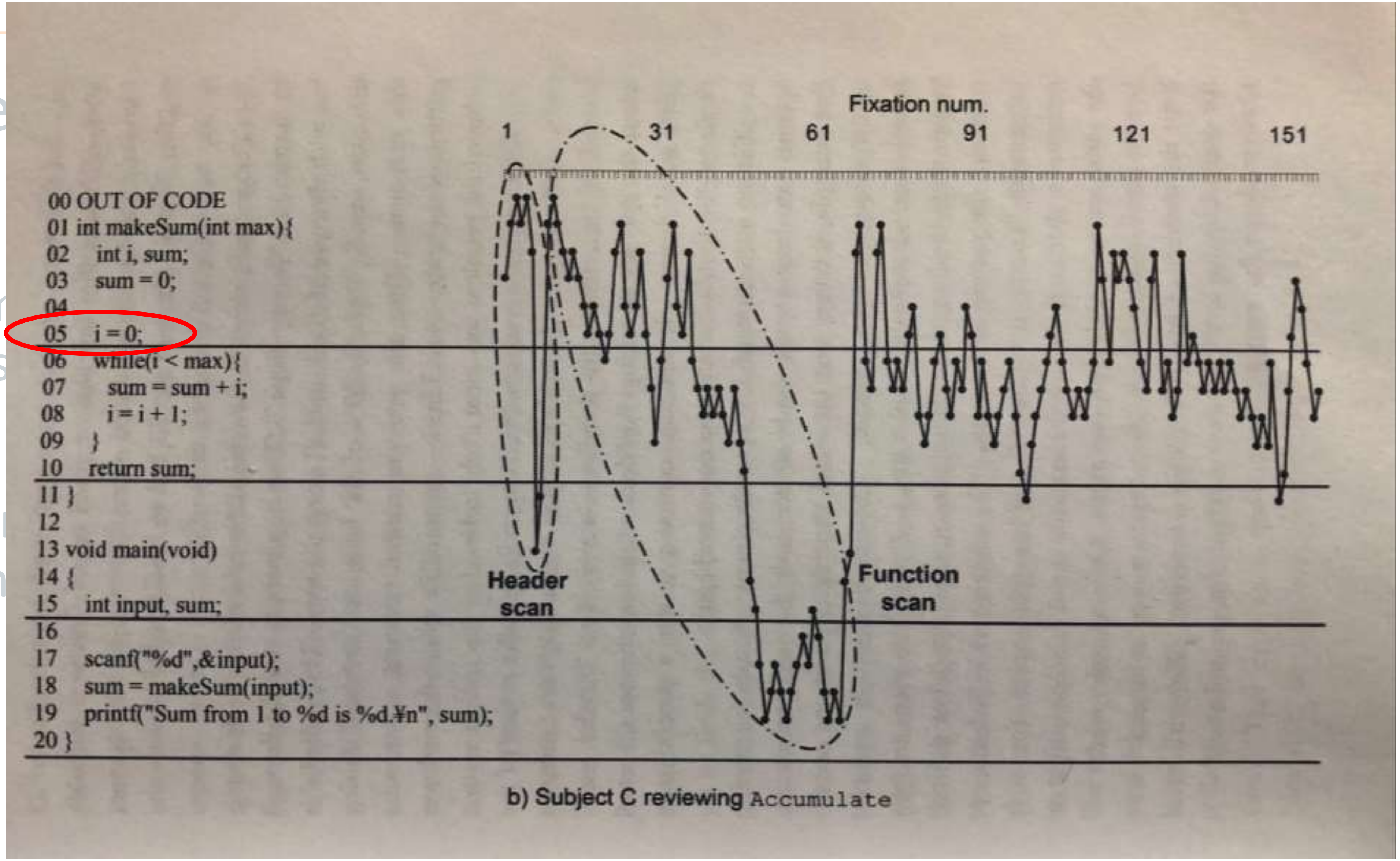
Research by Uwano at Nara Institute of Science and Technology. (2006)

The Best Kept Secrets of Peer Code Review

Revie

- Lim ses

- Co len



```
00 OUT OF CODE
01 int makeSum(int max){
02     int i, sum;
03     sum = 0;
04
05     i = 0;
06     while(i < max){
07       sum = sum + i;
08       i = i + 1;
09     }
10     return sum;
11 }
12
13 void main(void)
14 {
15     int input, sum;
16
17     scanf("%d",&input);
18     sum = makeSum(input);
19     printf("Sum from 1 to %d is %d.¥n", sum);
20 }
```

b) Subject C reviewing Accumulate

Research by Uwano at Nara Institute of Science and Technology. (2006)

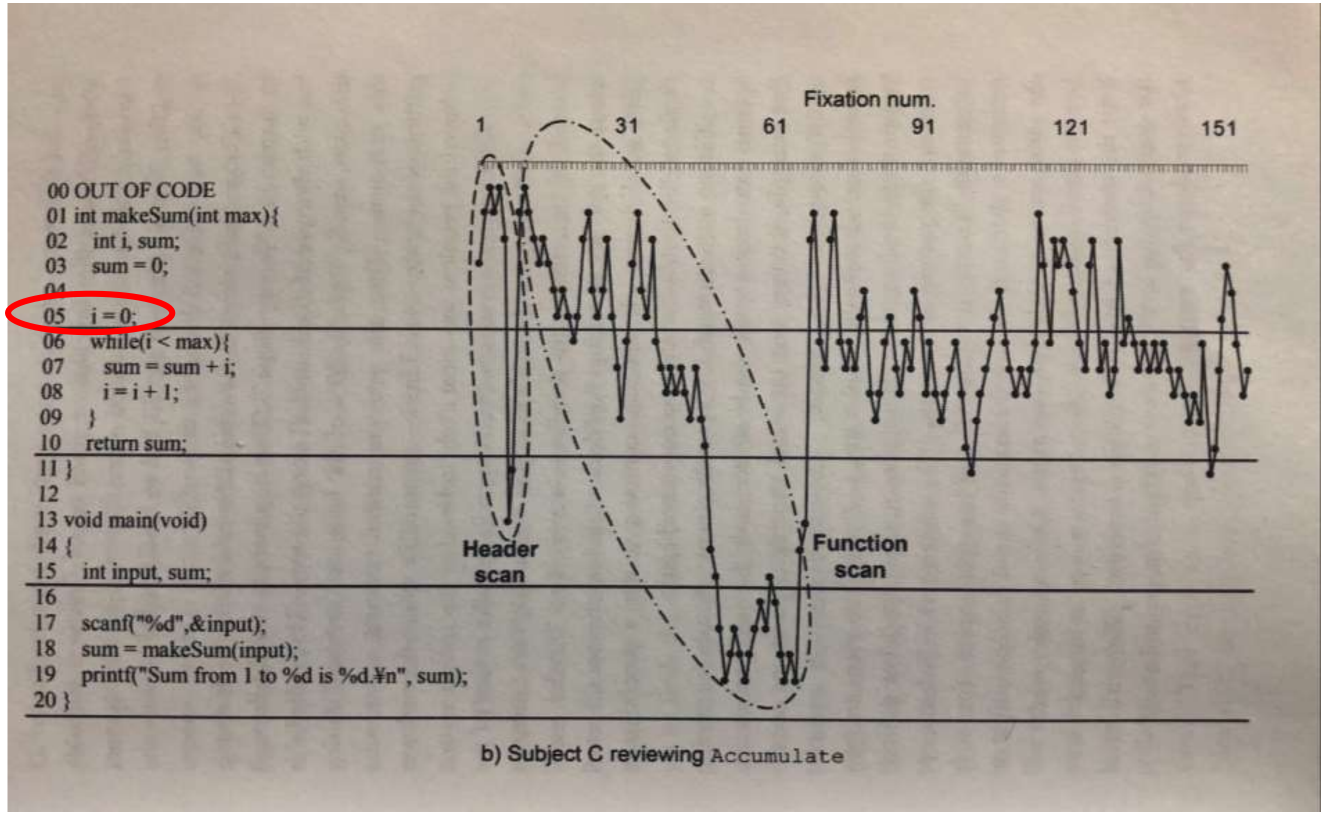
The Best Kept Secrets of Peer Code Review

SMARTBEAR

Review Guidelines

- Limit reviews to 60 minute sessions at most

- Consider shorter function length to improve readability



```
00 OUT OF CODE
01 int makeSum(int max){
02    int i, sum;
03    sum = 0;
04
05    i = 0;
06    while(i < max){
07      sum = sum + i;
08      i = i + 1;
09    }
10    return sum;
11 }
12
13 void main(void)
14 {
15    int input, sum;
16
17    scanf("%d",&input);
18    sum = makeSum(input);
19    printf("Sum from 1 to %d is %d.¥n", sum);
20 }
```

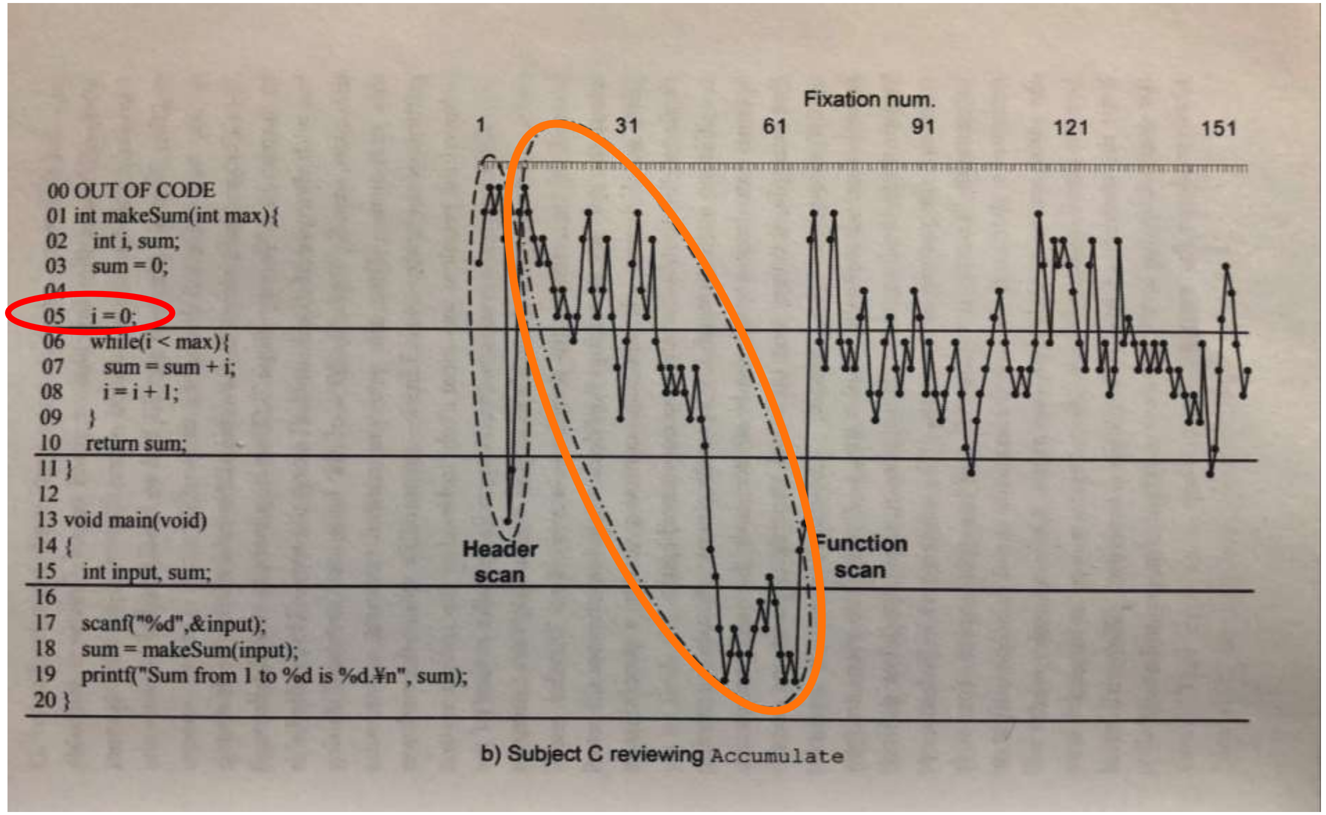
b) Subject C reviewing Accumulate

Research by Uwano at Nara Institute of Science and Technology. (2006)

The Best Kept Secrets of Peer Code Review

Review Guidelines

- Limit reviews to 60 minute sessions at most
- Consider shorter function length to improve readability
- Slow down your reading and take time to scan



```
int makeSum(int max){
  int i, sum;
  sum = 0;

  i = 0;
  while(i < max){
    sum = sum + i;
    i = i + 1;
  }
  return sum;
}

void main(void)
{
  int input, sum;

  scanf("%d",&input);
  sum = makeSum(input);
  printf("Sum from 1 to %d is %d.¥n", sum);
}
```

b) Subject C reviewing `Accumulate`

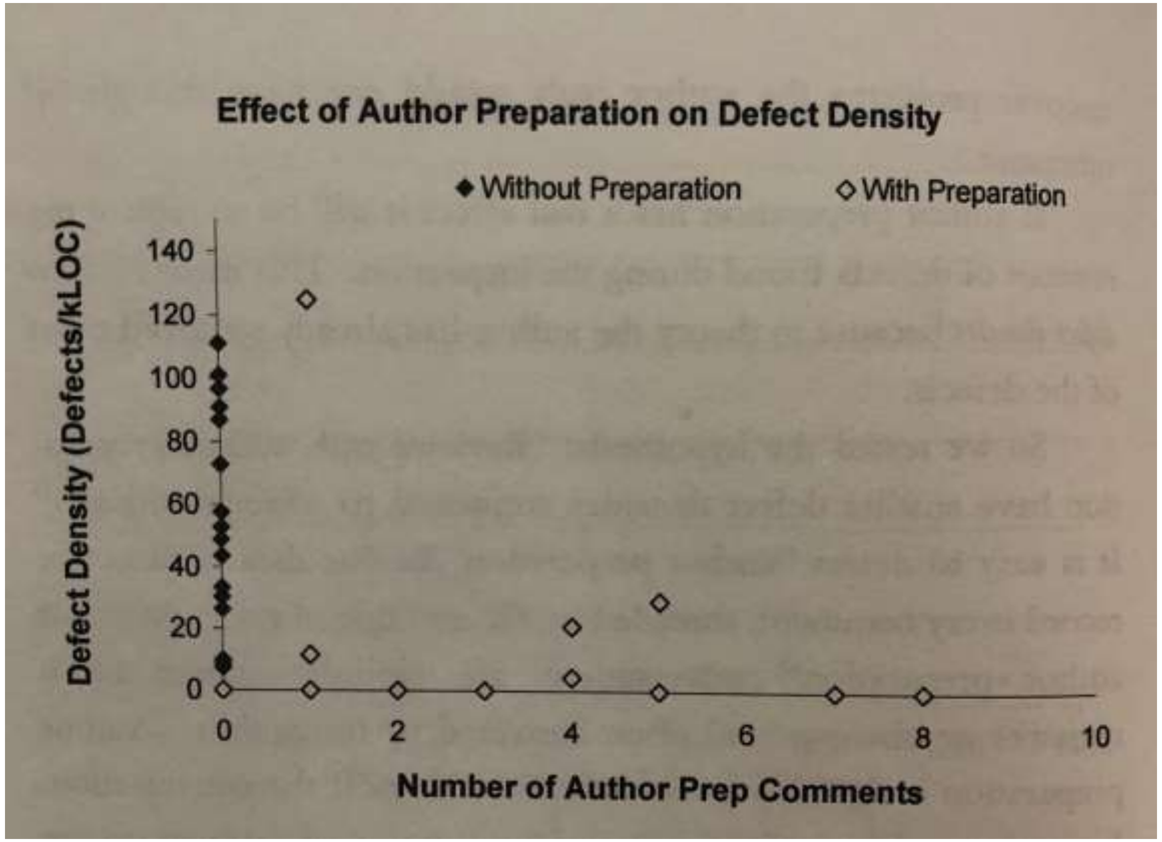
Research by Uwano at Nara Institute of Science and Technology. (2006)

The Best Kept Secrets of Peer Code Review

SMARTBEAR

Ideal Review

- Author annotates their code prior to reviewers reading



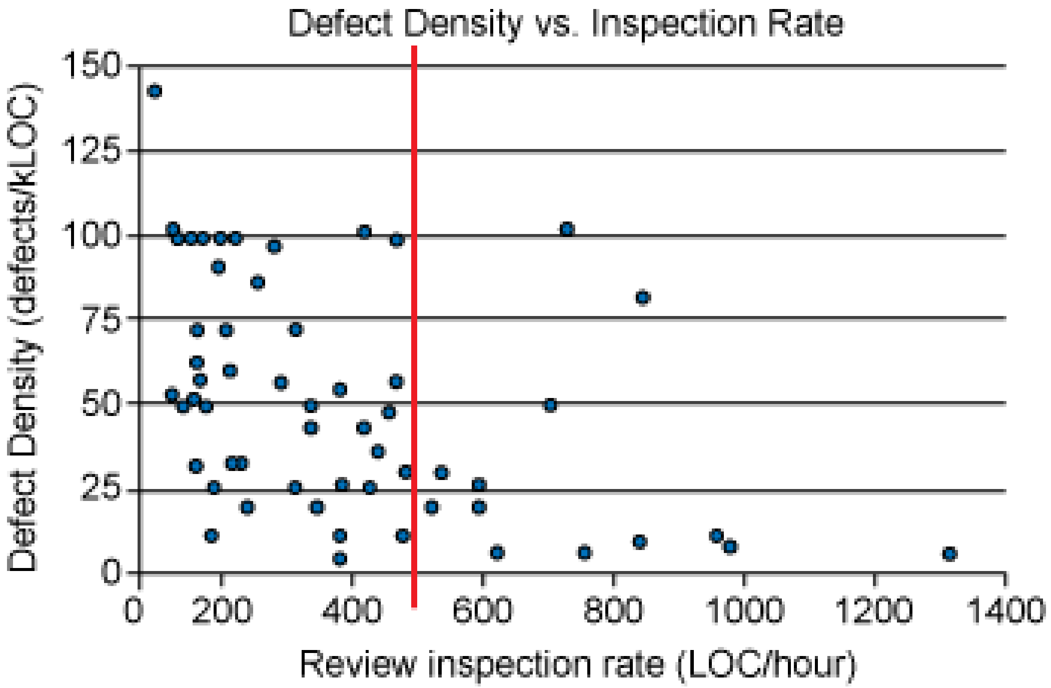
Cisco Case Study with SmartBear. (2006)

The Best Kept Secrets of Peer Code Review

Ideal Review

- Author annotates their code prior to reviewers reading

- 100 – 300 lines of code

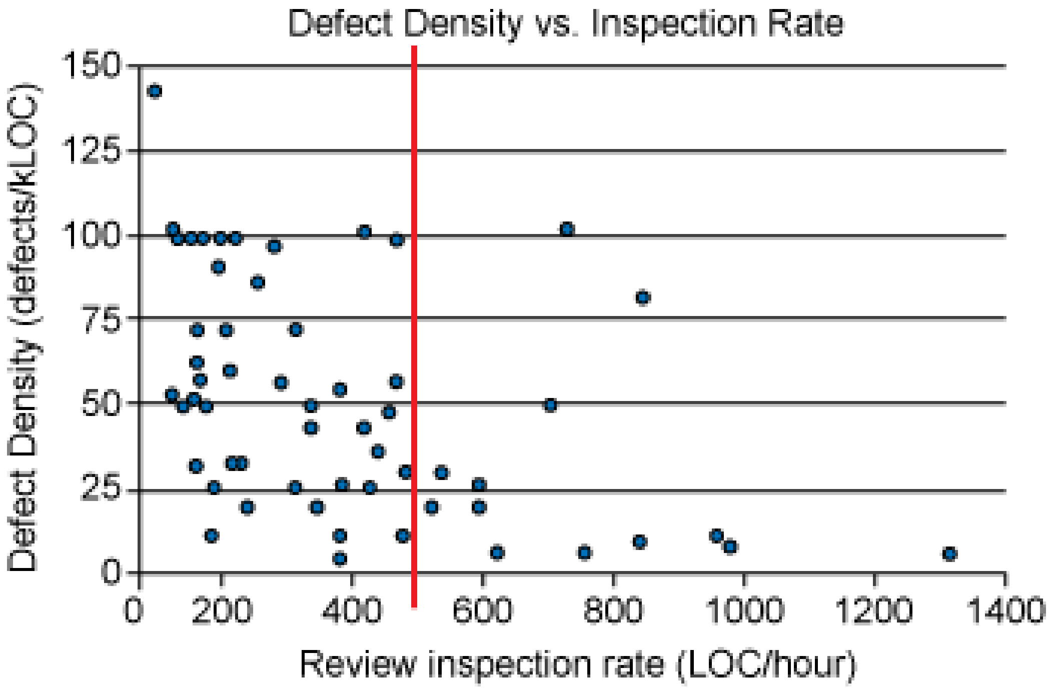


Cisco Case Study with SmartBear. (2006)

The Best Kept Secrets of Peer Code Review

Ideal Review

- Author annotates their code prior to reviewers reading
- 100 – 300 lines of code
- Reviewers spend 30-60 minutes reviewing the code change

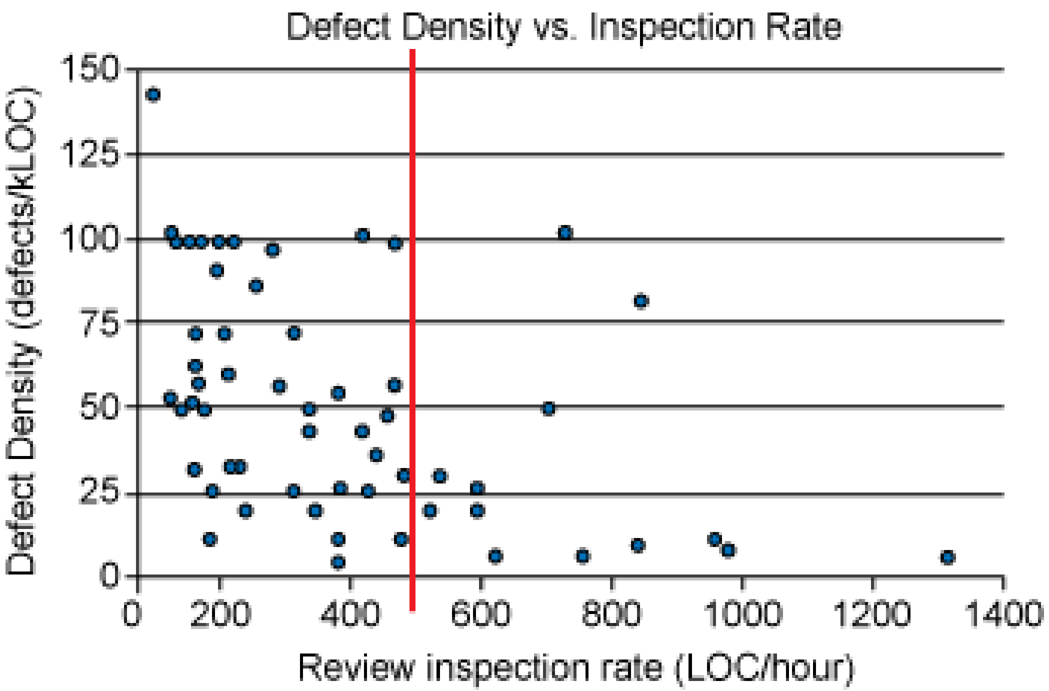


Cisco Case Study with SmartBear. (2006)

The Best Kept Secrets of Peer Code Review

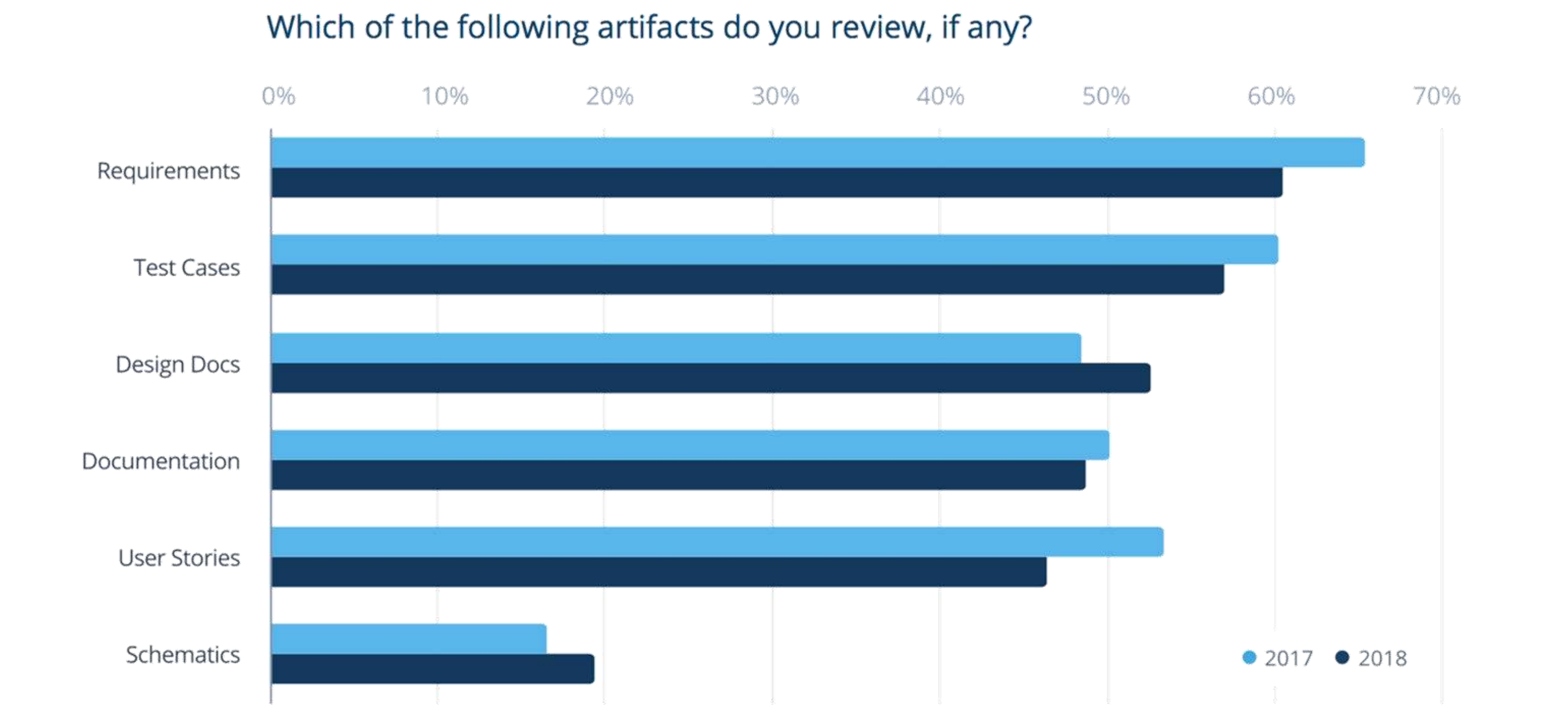
Goals & Reading Metrics

- Internal Metrics
  - Inspection Rate
  - Defect Rate
  - Defect Density
  - Defect Count
  - Defect Type and Severity



- External Metrics
  - Number of support calls
  - Number of bugs injected by Development

Reviewing Artifacts



Reviewing Artifacts

90% of teams are conducting document review of some kind.

35% of respondents are using a tool to do this.

Teams using a tool for reviewing documents are

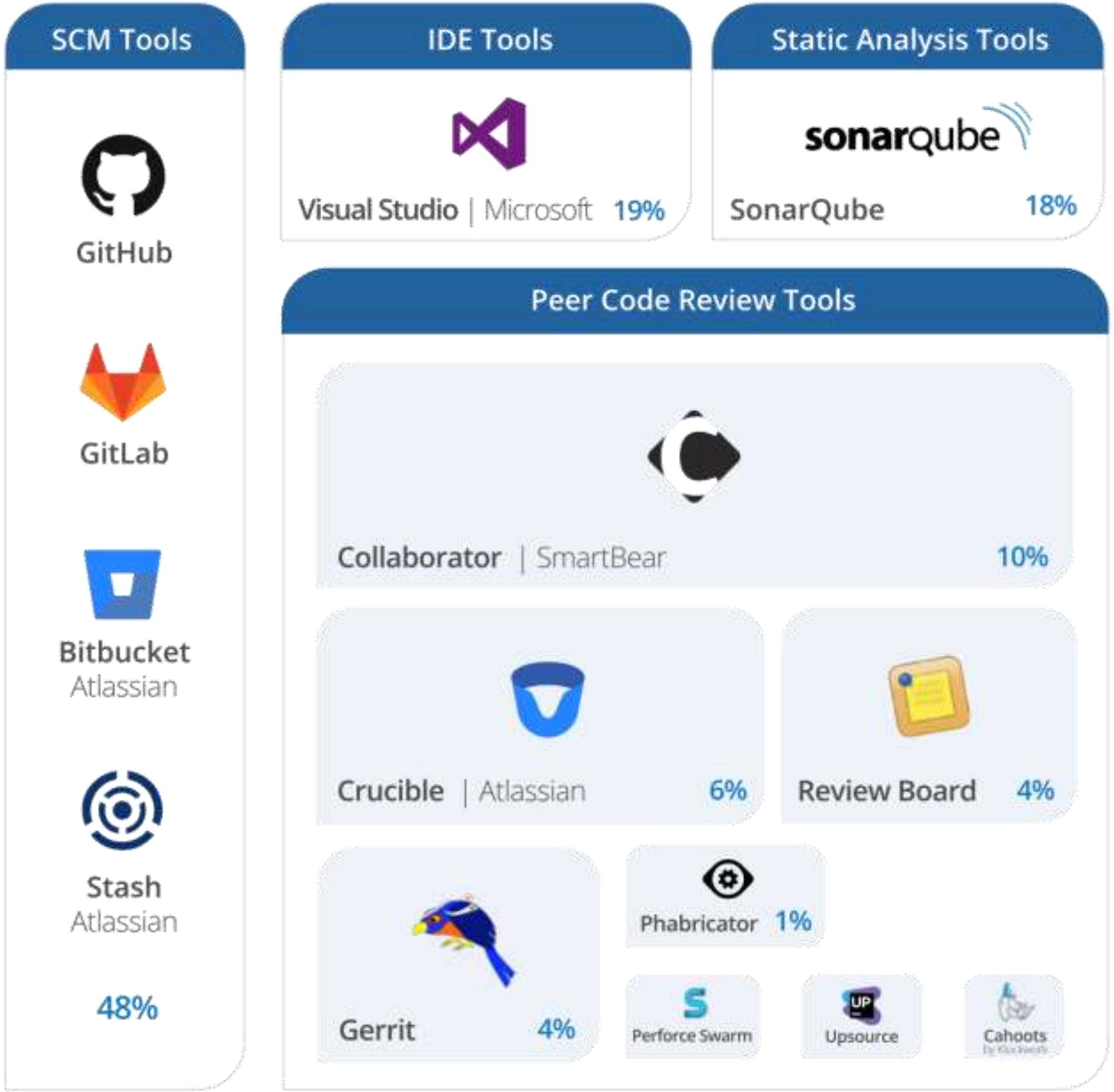
26% more likely

to be satisfied with their code review process.

Review Best Practices

- Conduct reviews daily
- Set clear expectations with checklists
- Conduct tool-assisted reviews + meetings
- Create goals and report on your process
- Foster a learning culture
- Limit reviews to 60 minute sessions
- Shorten function length for readability
- Slow down your reading and take time to scan
- Author annotates code prior to adding reviewers
- 100 – 300 lines of code at a time
- Reviewers should spend 30-60 minutes at a time
- Verify that defects have been addressed
- Conduct peer reviews on other docs & artifacts
- Maintain review history for long-term success

Current Review Tool Landscape



SCM Tools

GitHub

GitLab

Bitbucket
Atlassian

Stash
Atlassian

48%

IDE Tools

Visual Studio | Microsoft 19%

Static Analysis Tools

sonarqube

SonarQube 18%

Peer Code Review Tools

Collaborator | SmartBear 10%

Crucible | Atlassian 6%

Review Board 4%

Gerrit 4%

Phabricator 1%

Perforce Swarm

Upsource

Cahoots
by Klocwork

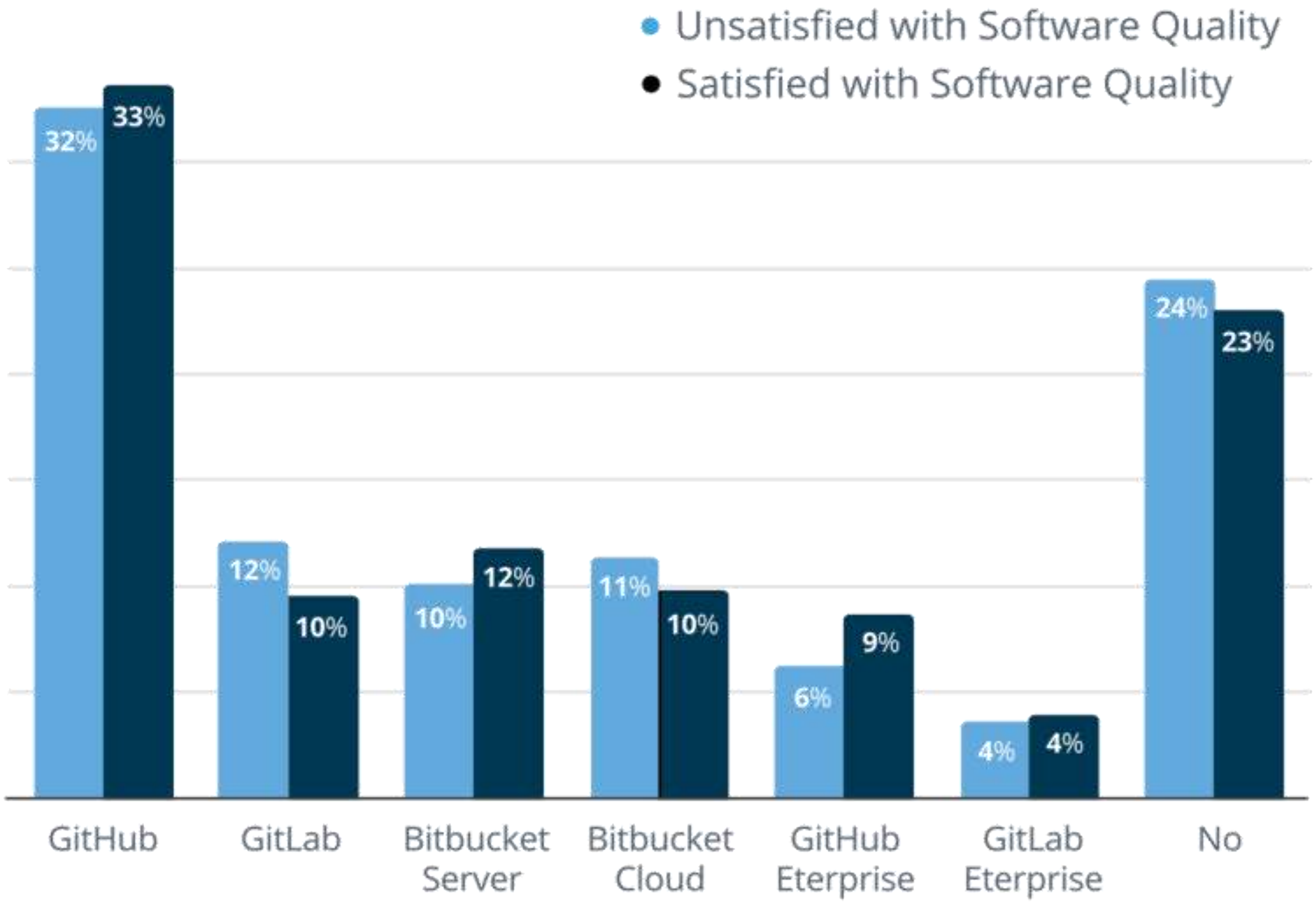
Review Best Practices

- Conduct reviews daily
- Set clear expectations with checklists
- Conduct tool-assisted reviews + meetings
- Create goals and report on your process
- Foster a learning culture
- Limit reviews to 60 minute sessions
- Shorten function length for readability
- Slow down your reading and take time to scan
- Author annotates code prior to adding reviewers
- 100 – 300 lines of code at a time
- Reviewers should spend 30-60 minutes at a time
- Verify that defects have been addressed
- Conduct peer reviews on other docs & artifacts
- Maintain review history for long-term success

Review Best Practices

- Conduct reviews daily
- Set clear expectations with checklists
- Conduct tool-assisted reviews + meetings
- Create goals and report on your process
- Foster a learning culture
- Limit reviews to 60 minute sessions
- Shorten function length for readability
- Slow down your reading and take time to scan
- Author annotates code prior to adding reviewers
- 100 – 300 lines of code at a time
- Reviewers should spend 30-60 minutes at a time
- Verify that defects have been addressed
- Conduct peer reviews on other docs & artifacts
- Maintain review history for long-term success

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



The 2018 State of Code Review

Review Best Practices

- **Conduct reviews daily**
- Set clear expectations with checklists
- **Conduct tool-assisted reviews + meetings**
- Create goals and report on your process
- **Foster a learning culture**
- **Limit reviews to 60 minute sessions**
- **Shorten function length for readability**

- **Slow down your reading and take time to scan**
- Author annotates code prior to adding reviewers
- **100 – 300 lines of code at a time**
- **Reviewers should spend 30-60 minutes at a time**
- Verify that defects have been addressed
- Conduct peer reviews on other docs & artifacts
- Maintain review history for long-term success

Smart Review Practices

- Set clear expectations with checklists
- Create goals and report on your process
- Author annotates code prior to adding reviewers
- Verify that defects have been addressed
- Conduct peer reviews on other docs & artifacts
- Maintain review history for long-term success

*Teams that report on their code reviews are **5X as likely to be satisfied** with their process*

*Teams that review documents in tool are **26% more likely to be satisfied** with their code reviews.*

Review Best Practices

SMARTBEAR
Collaborator

- Conduct reviews daily
- Set clear expectations with checklists
- Conduct tool-assisted reviews + meetings
- Create goals and report on your process
- Foster a learning culture
- Limit reviews to 60 minute sessions
- Shorten function length for readability

- Slow down your reading and take time to scan
- Author annotates code prior to adding reviewers
- 100 – 300 lines of code at a time
- Reviewers should spend 30-60 minutes at a time
- Verify that defects have been addressed
- Conduct peer reviews on other docs & artifacts
- Maintain review history for long-term success

Reviews in Collaborator

SMARTBEAR

Thank you!