Nasir Eisty
Assistant Professor
Dept. of CSSE
Cal Poly State University
San Luis Obispo, California

Testing and Code Review Practices in Research Software

Who Am I?

LEARN BY DOING
CAL POLY
Computer Science &
Software Engineering
COLLEGE OF ENGINEERING

College of
Engineering
Computer Science

better
scientific
software

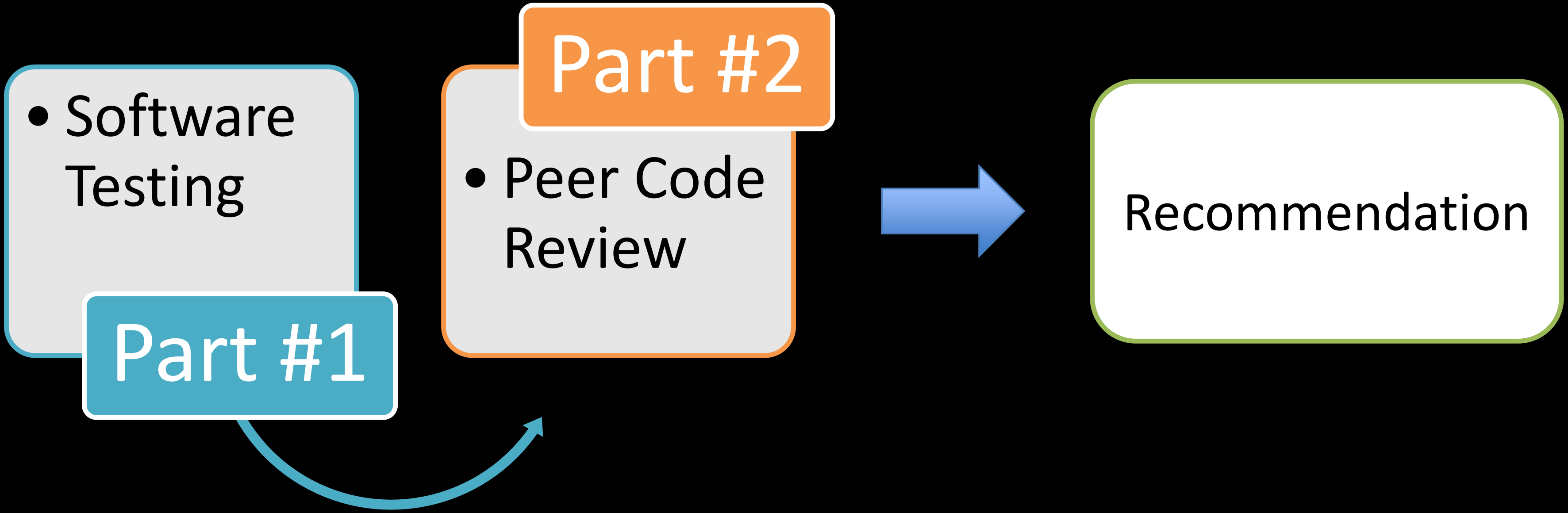
NCSA

Los Alamos
NATIONAL LABORATORY

Research Software

Software Quality

Contents



- Software Testing

Part #1

Part #2

- Peer Code Review

Recommendation

Part #1

Testing Research Software: A survey

Online Survey

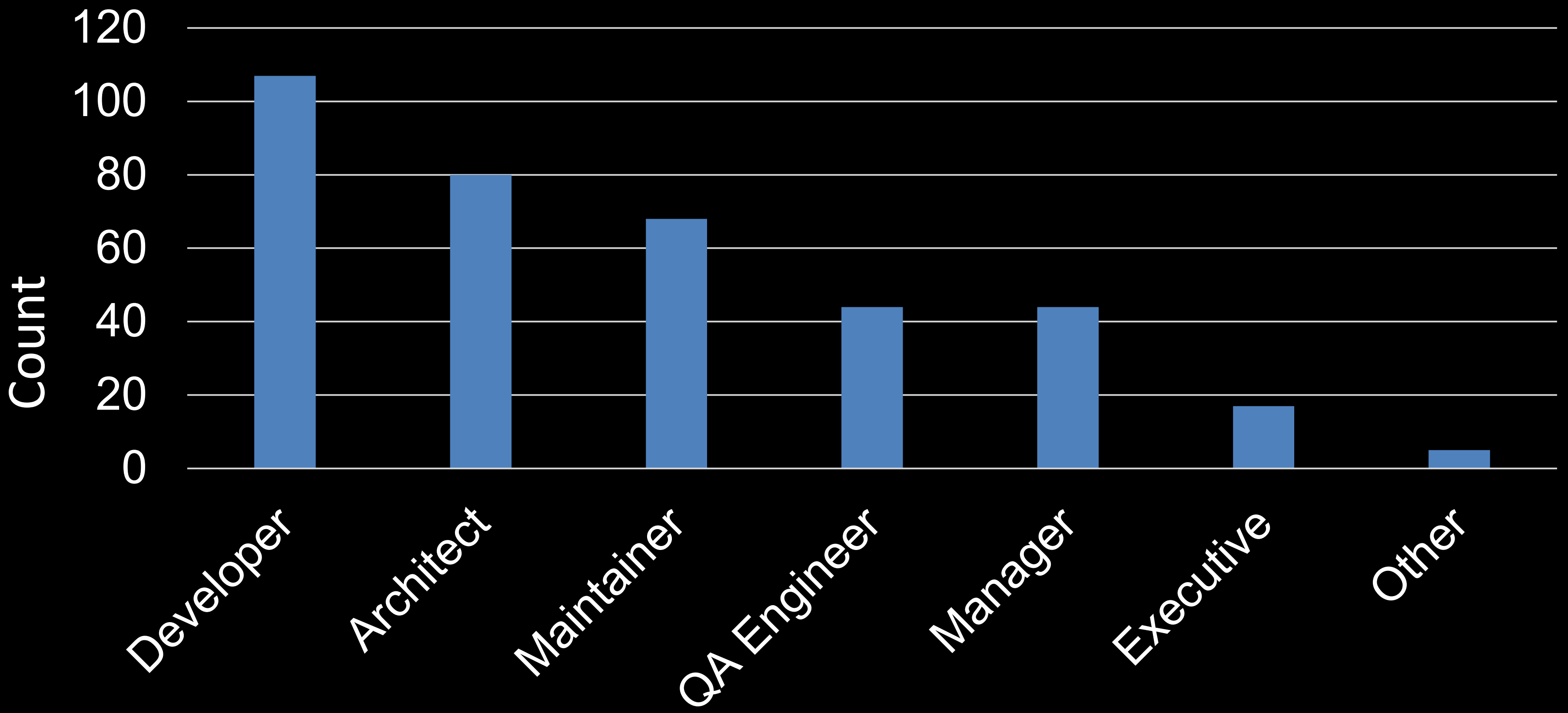
Part #1 (Testing) Outline

- Demographics
- Level of knowledge research software developers have on testing
- Current testing practices in research software community
- Difficulties to test research software
- Compatibility of Commercial/IT testing techniques
- Improvement of the testing process in research software

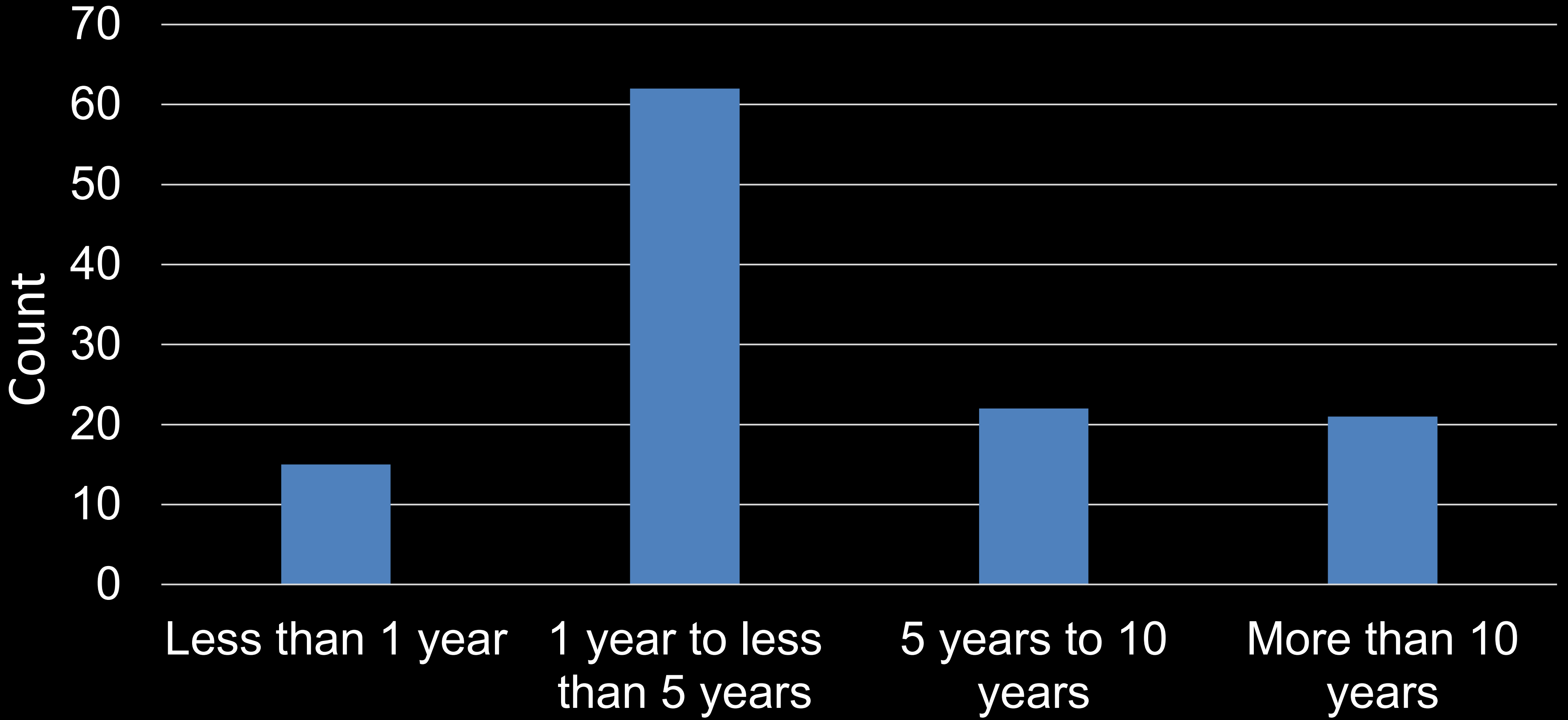
Part #1 (Testing) Outline

- Demographics
- Level of knowledge research software developers have on testing
- Current testing practices in research software community
- Difficulties to test research software
- Compatibility of Commercial/IT testing techniques
- Improvement of the testing process in researach software

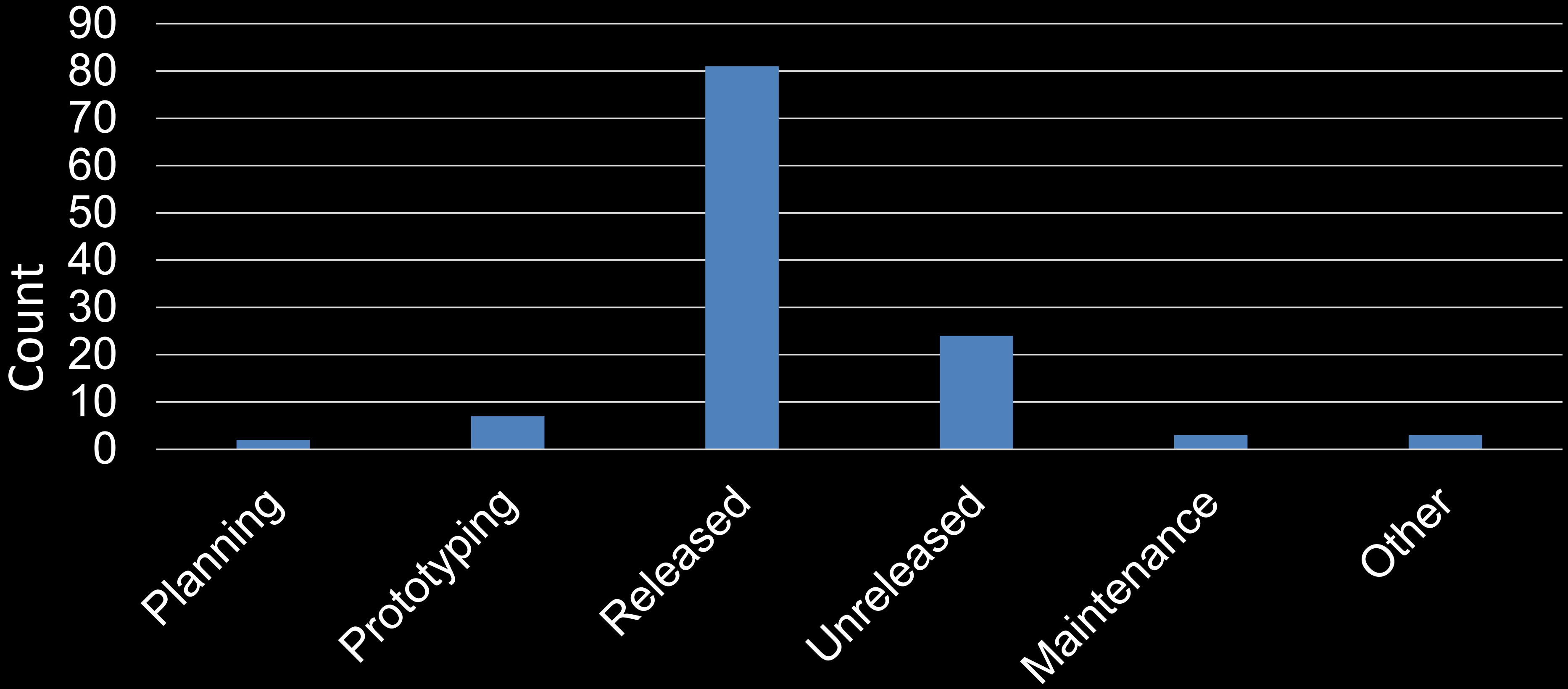
Roles


Count
120
100
80
60
40
20
0
Developer
Architect
Maintainer
QA Engineer
Manager
Executive
Other

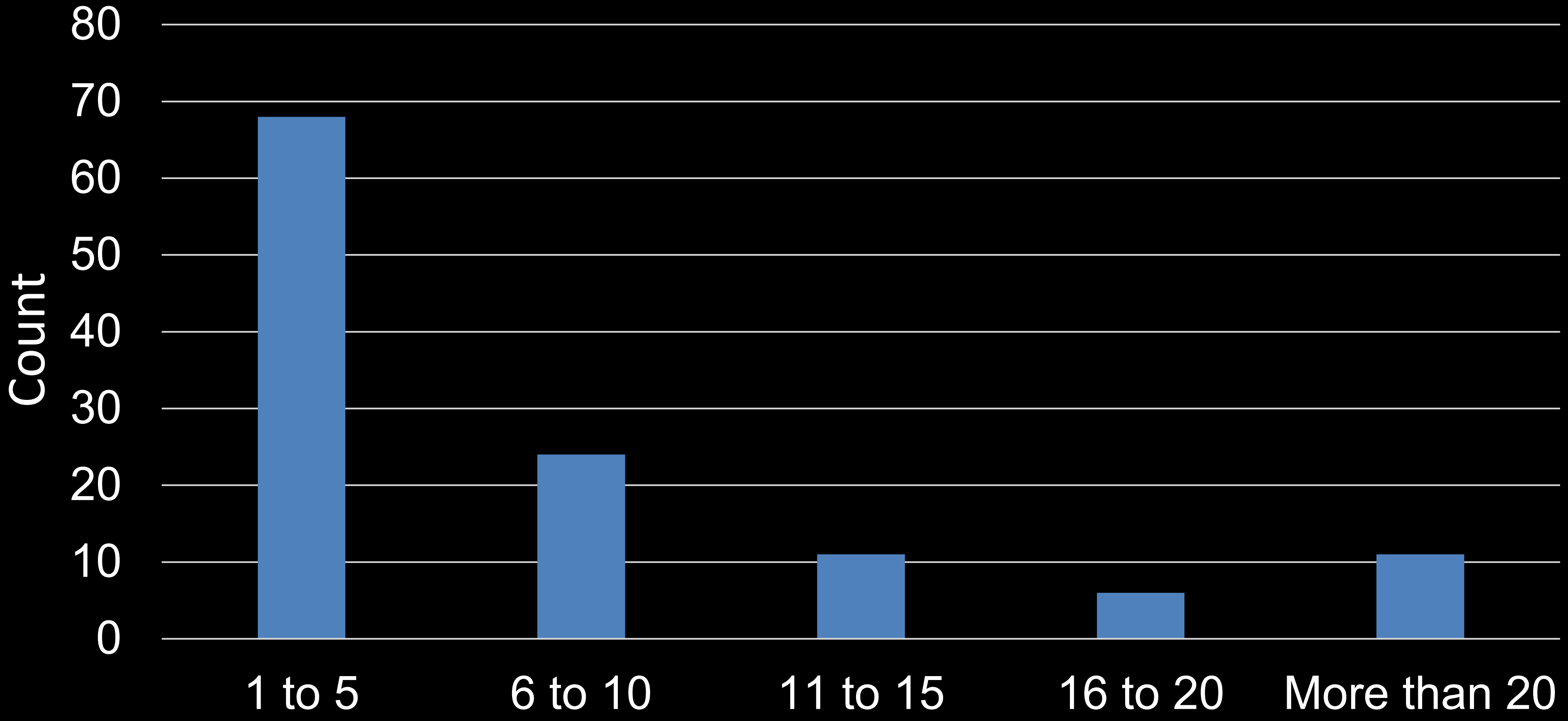
Years Worked



Project Stage



# Developers

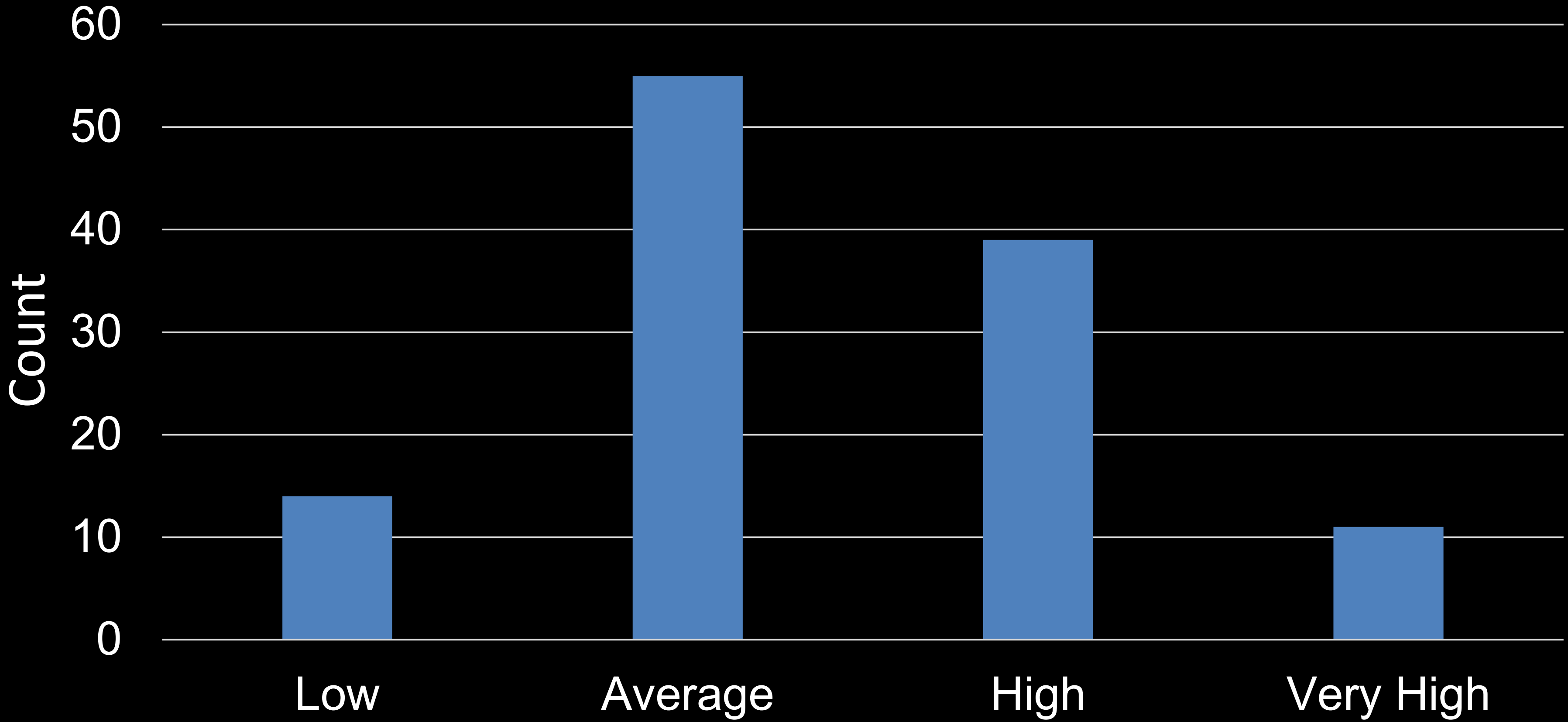


Count
80
70
60
50
40
30
20
10
0
1 to 5
6 to 10
11 to 15
16 to 20
More than 20

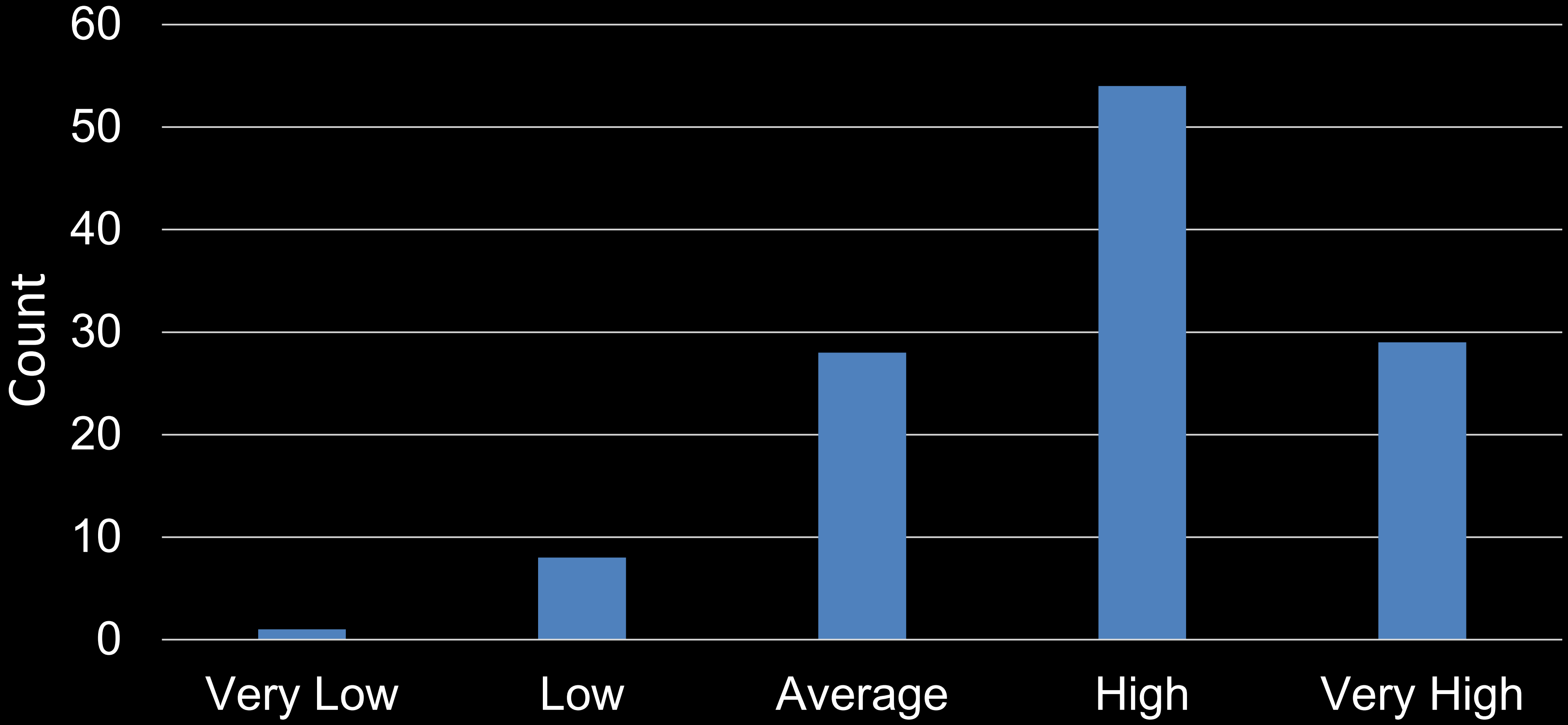
Part #1 (Testing) Outline

- Demographics
- Level of knowledge research software developers have on testing
- Current testing practices in research software community
- Difficulties to test research software
- Compatibility of Commercial/IT testing techniques
- Improvement of the testing process in research software

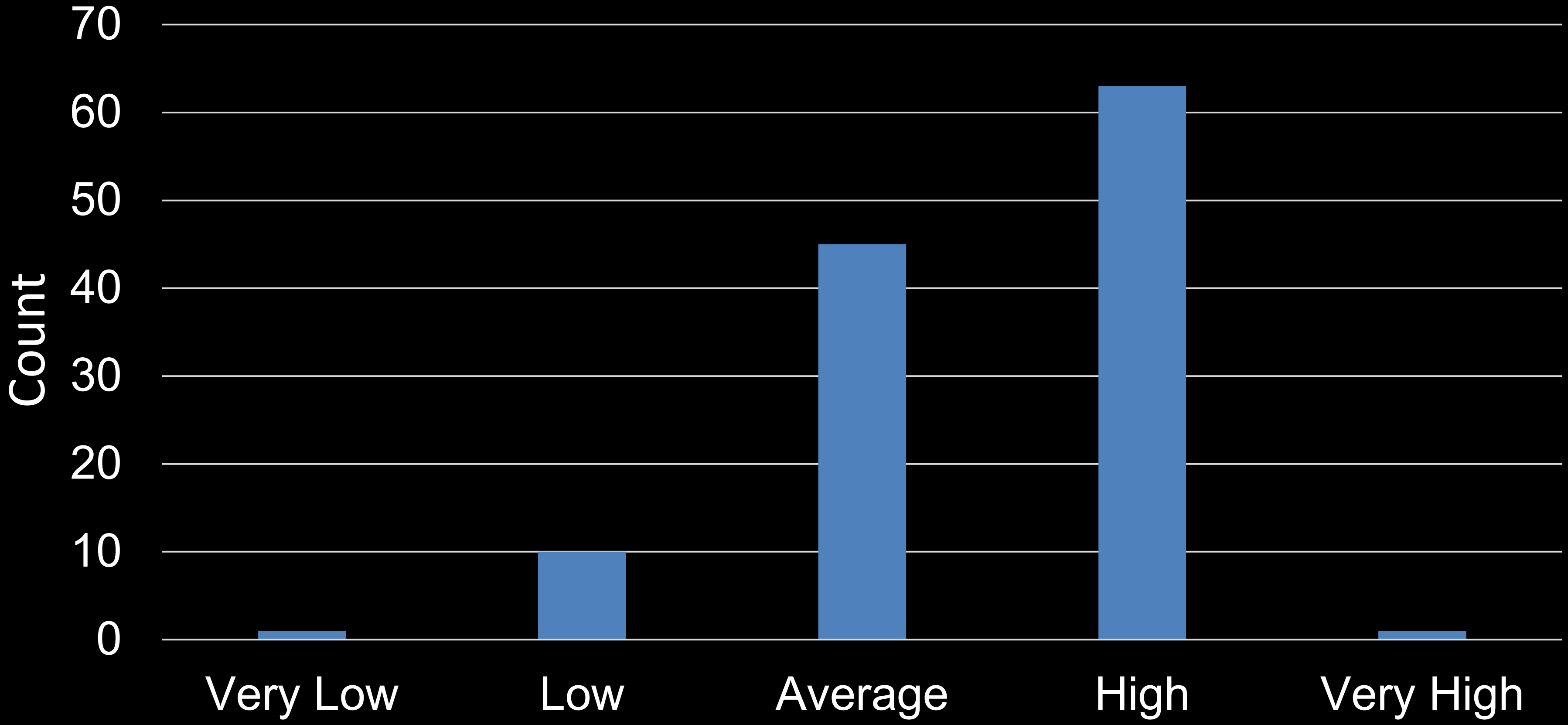
Knowledge of Testing



Understanding Testing Concepts USED



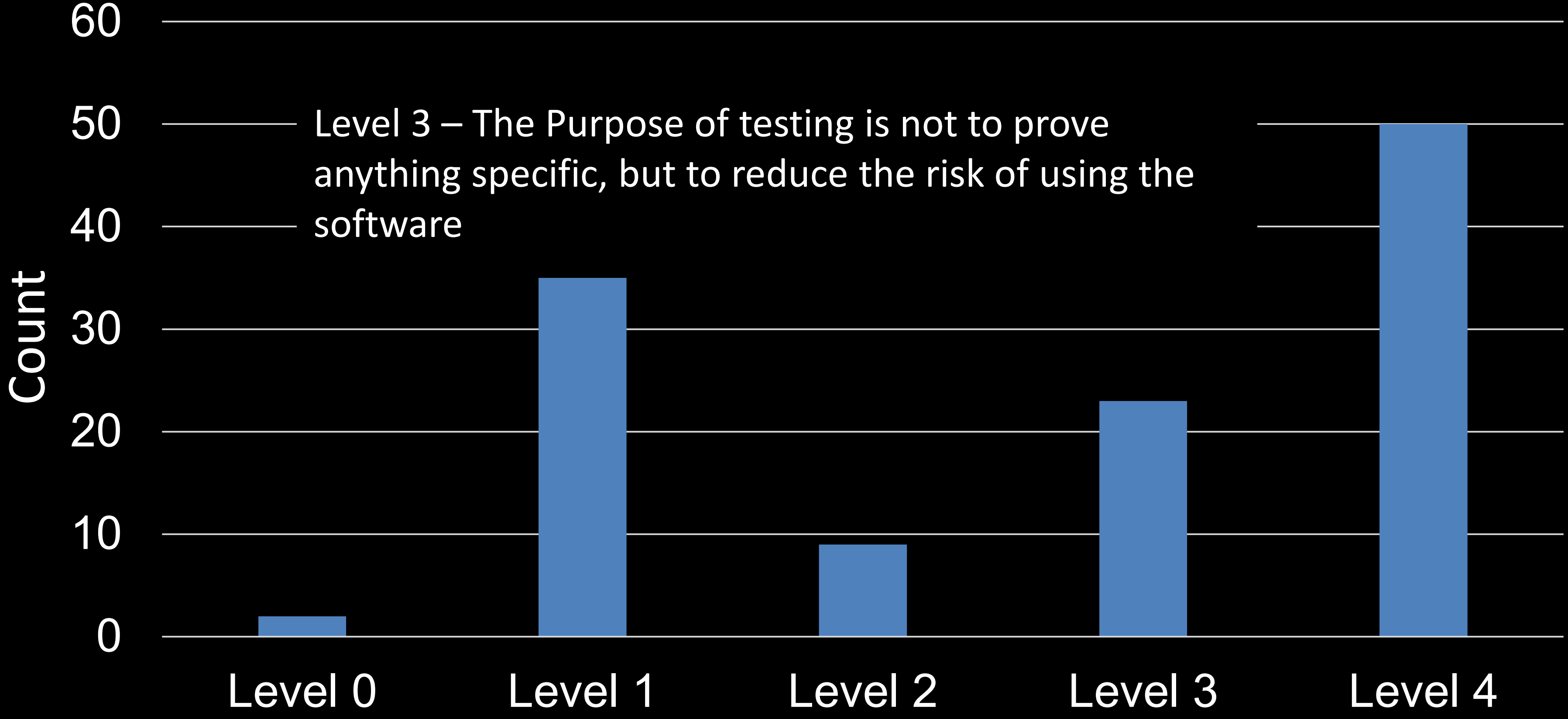
Understanding Testing Concepts NEEDED



Part #1 (Testing) Outline

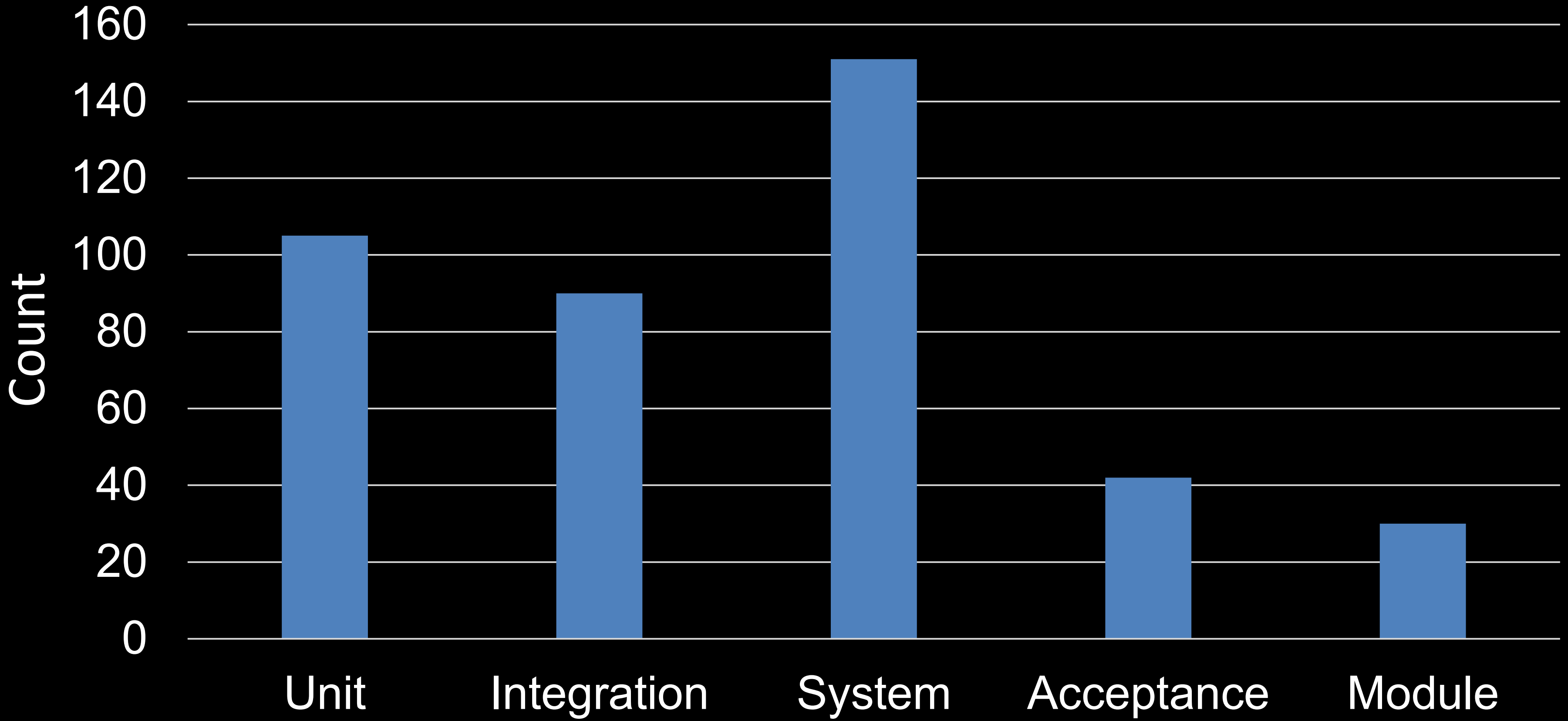
- Demographics
- Level of knowledge research software developers have on testing
- Current testing practices in research software community
- Difficulties to test research software
- Compatibility of Commercial/IT testing techniques
- Improvement of the testing process in research software

Testing Goals

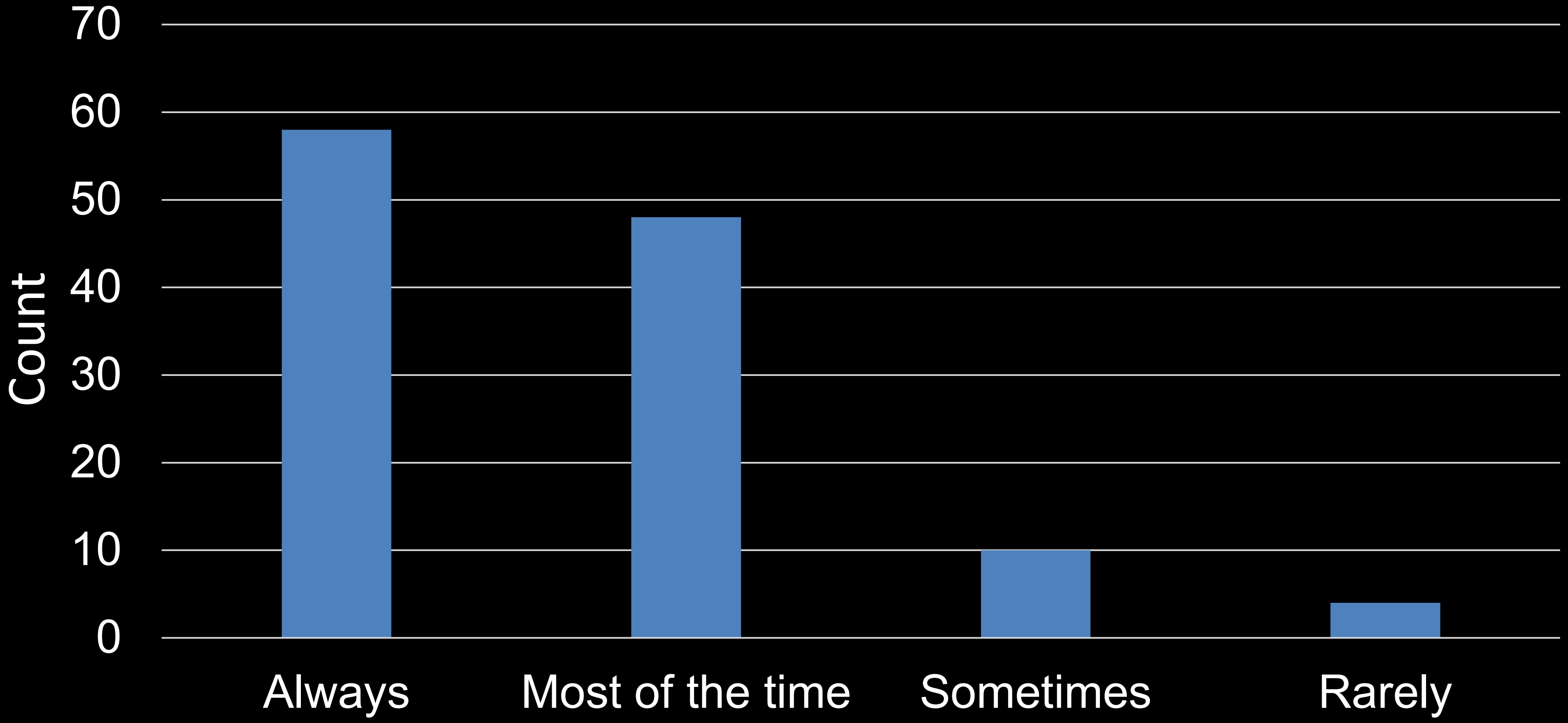


Level 3 – The Purpose of testing is not to prove anything specific, but to reduce the risk of using the software

Testing Methods Used



Usefulness

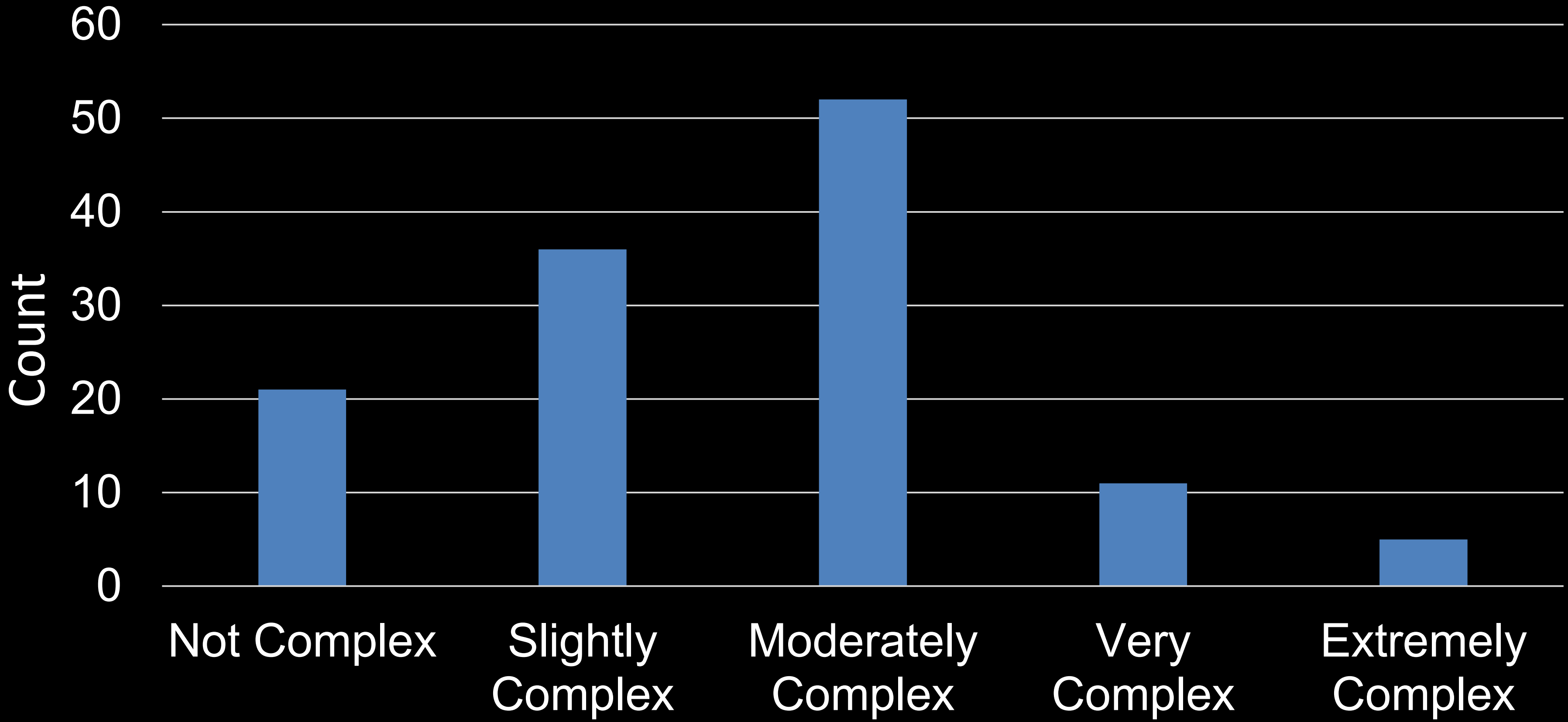


Count
70
60
50
40
30
20
10
0
Always
Most of the time
Sometimes
Rarely

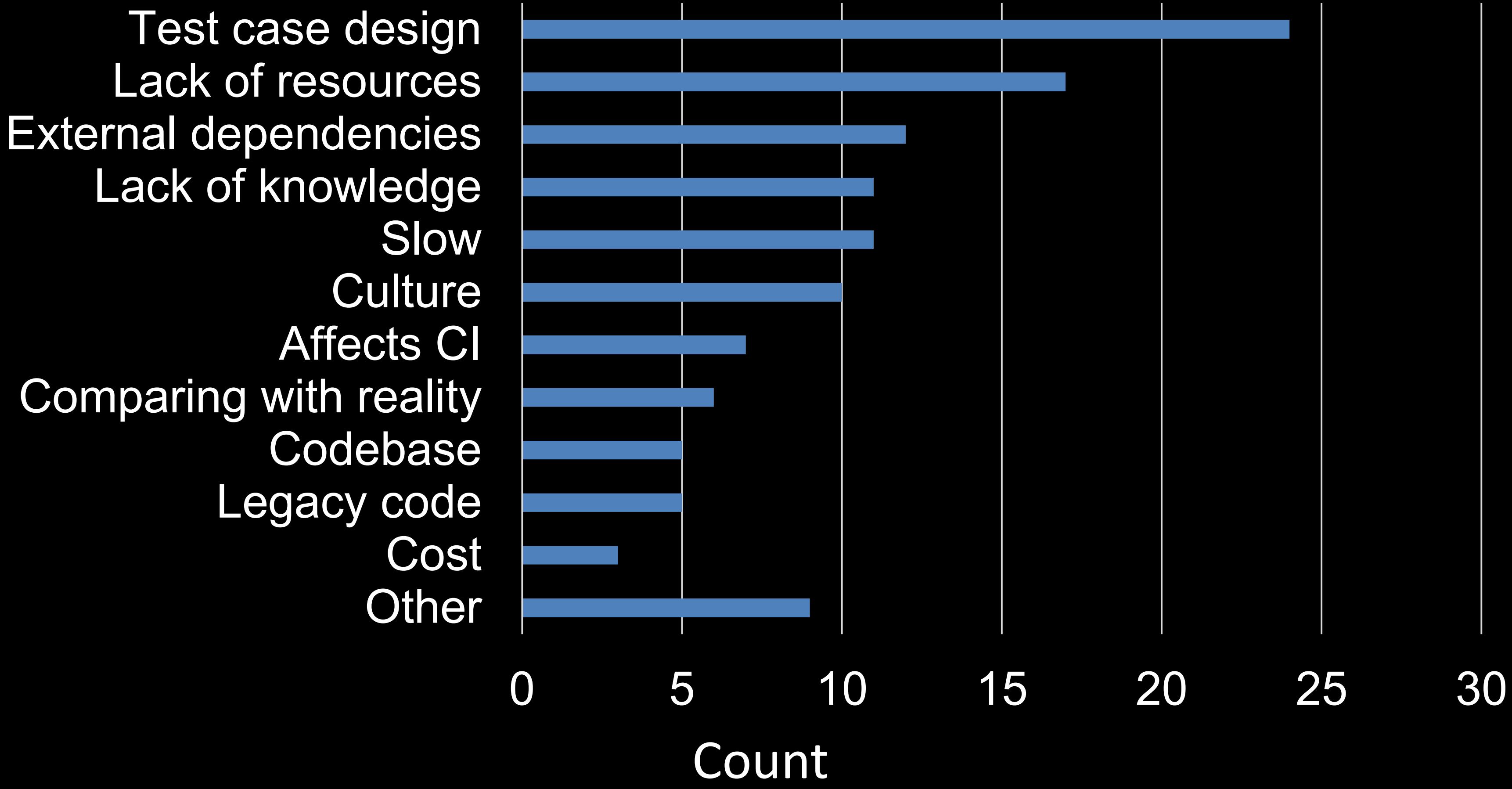
Part #1 (Testing) Outline

- Demographics
- Level of knowledge research software developers have on testing
- Current testing practices in research software community
- Difficulties to test research software
- Compatibility of Commercial/IT testing techniques
- Improvement of the testing process in research software

Complexity to Test



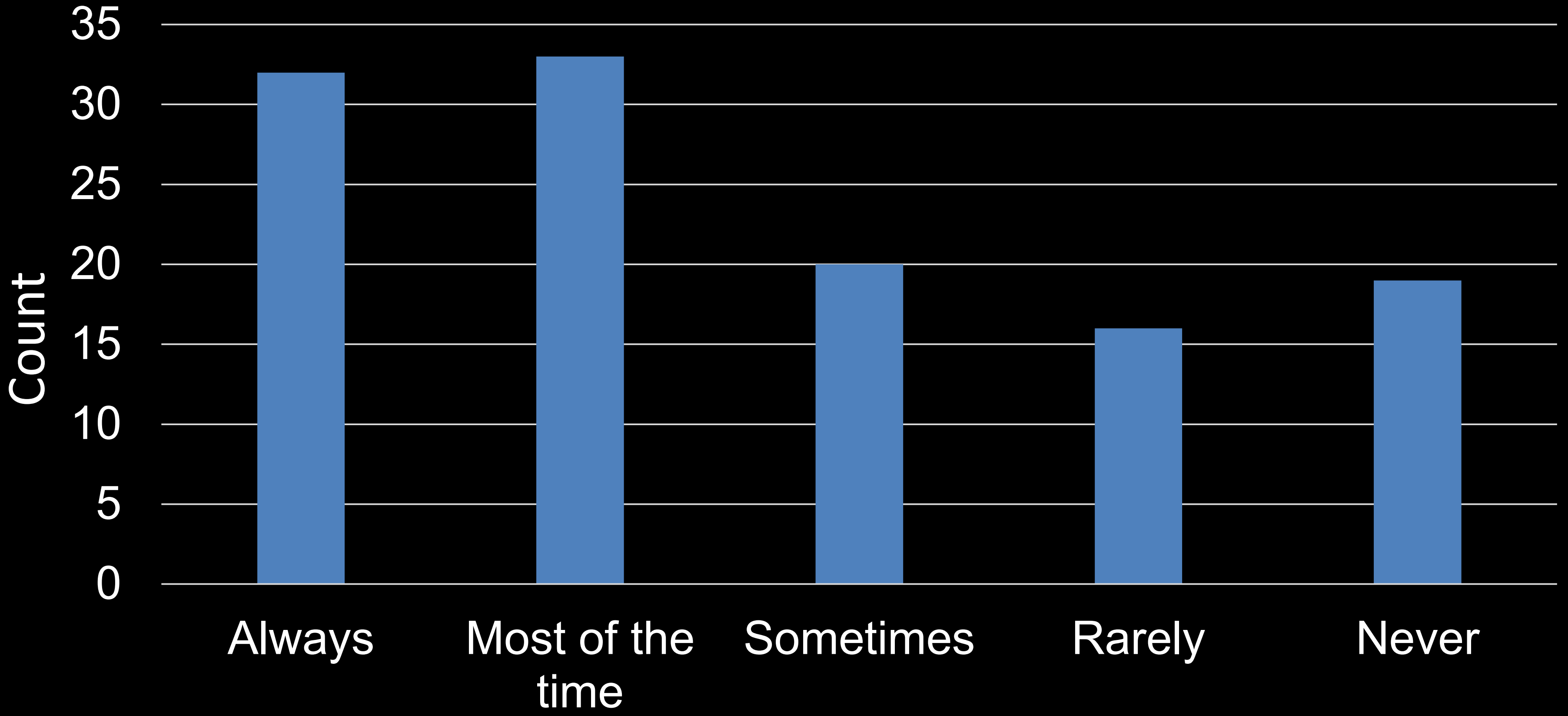
Challenges


Test case design
Lack of resources
External dependencies
Lack of knowledge
Slow
Culture
Affects CI
Comparing with reality
Codebase
Legacy code
Cost
Other
0
5
10
15
20
25
30
Count

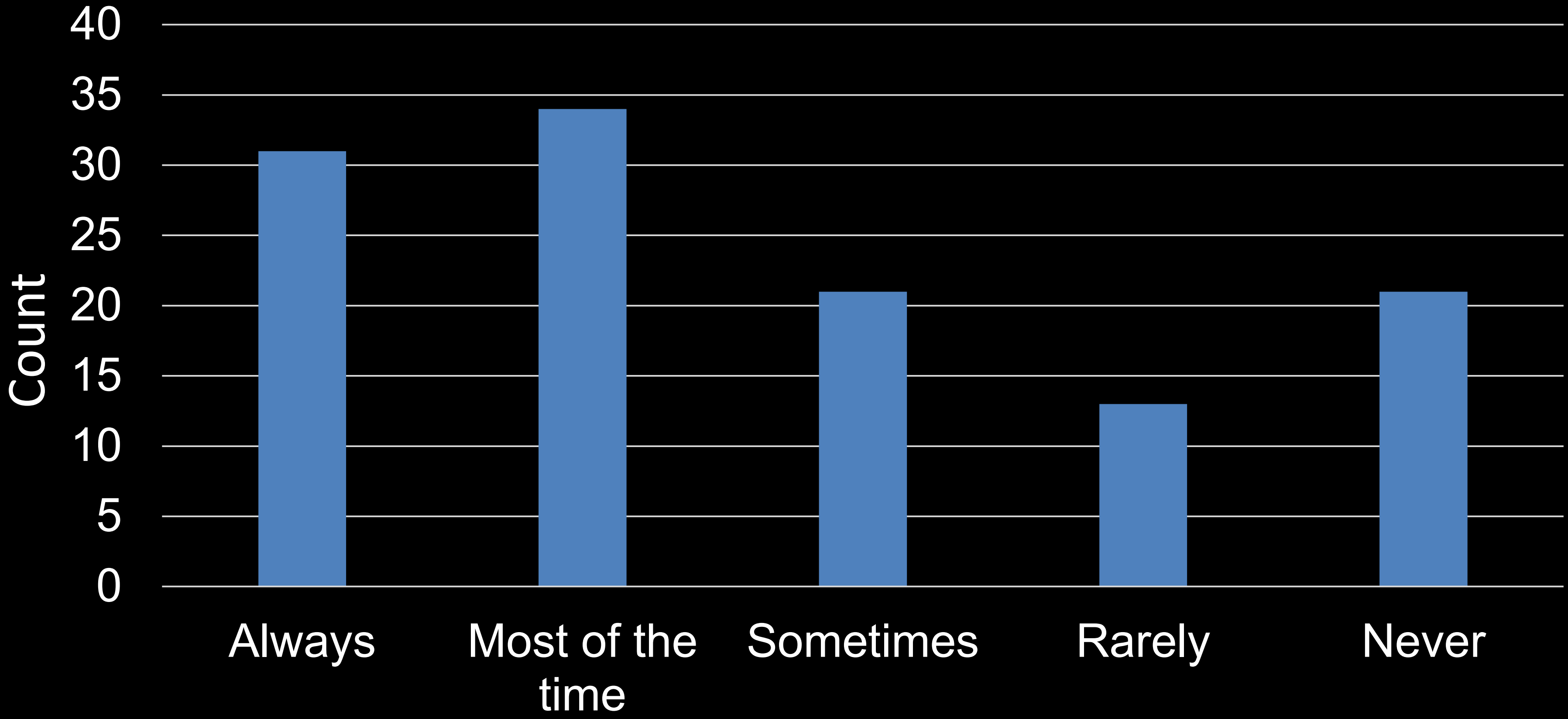
Part #1 (Testing) Outline

- Demographics
- Level of knowledge research software developers have on testing
- Current testing practices in research software community
- Difficulties to test research software
- Compatibility of Commercial/IT testing techniques
- Improvement of the testing process in research software

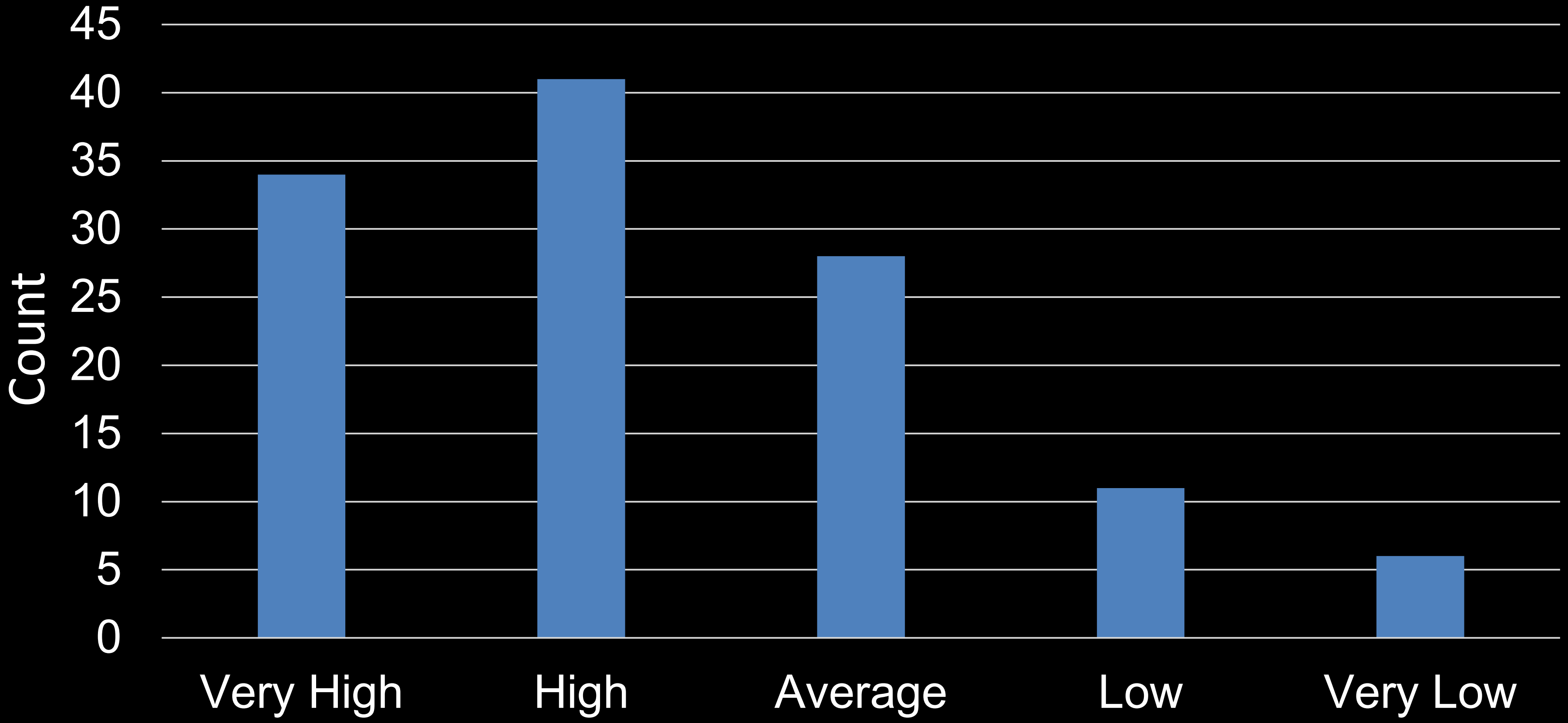
Commercial/IT Testing Methods - Team



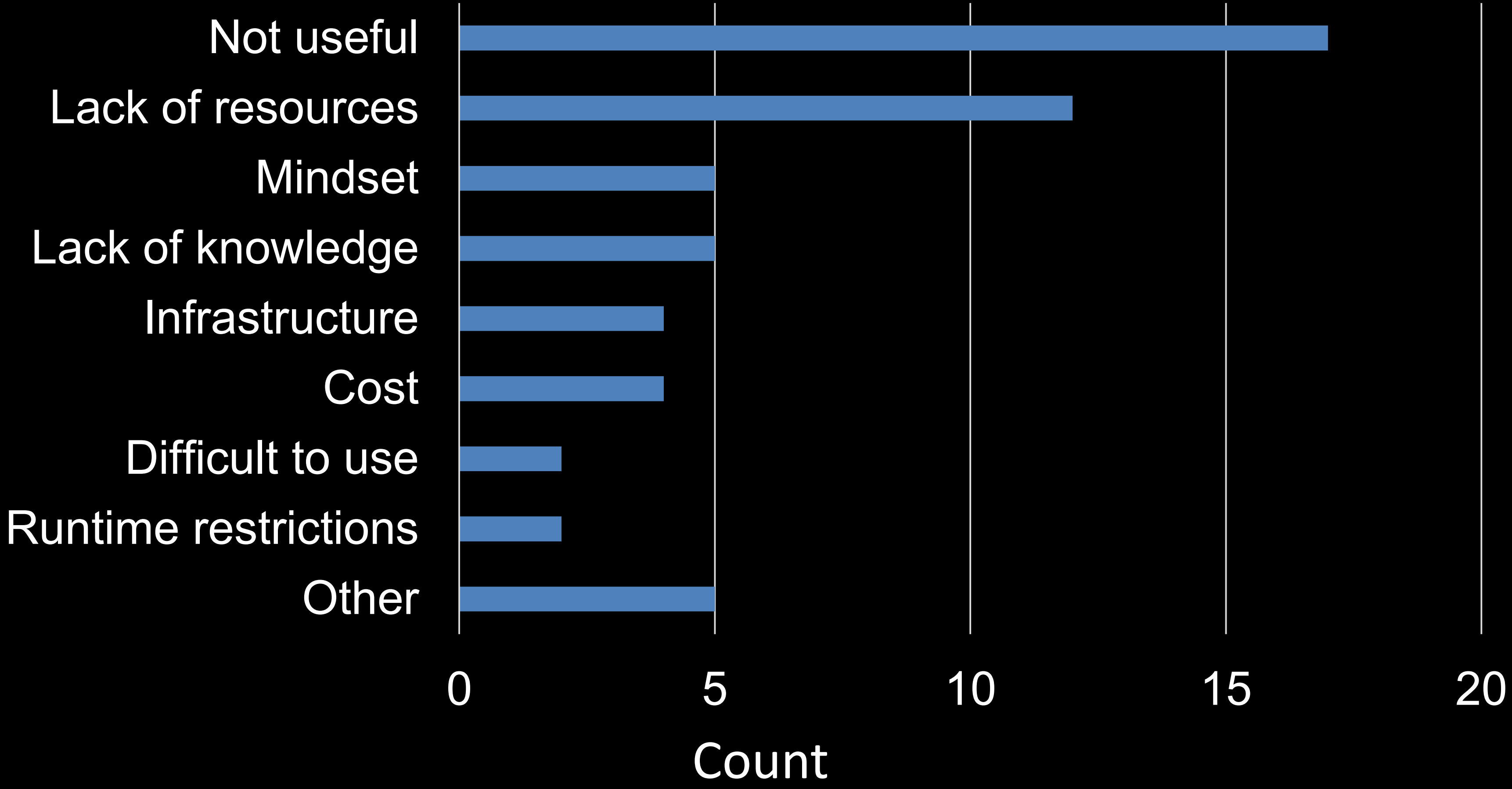
Commercial/IT Testing Methods- Individual



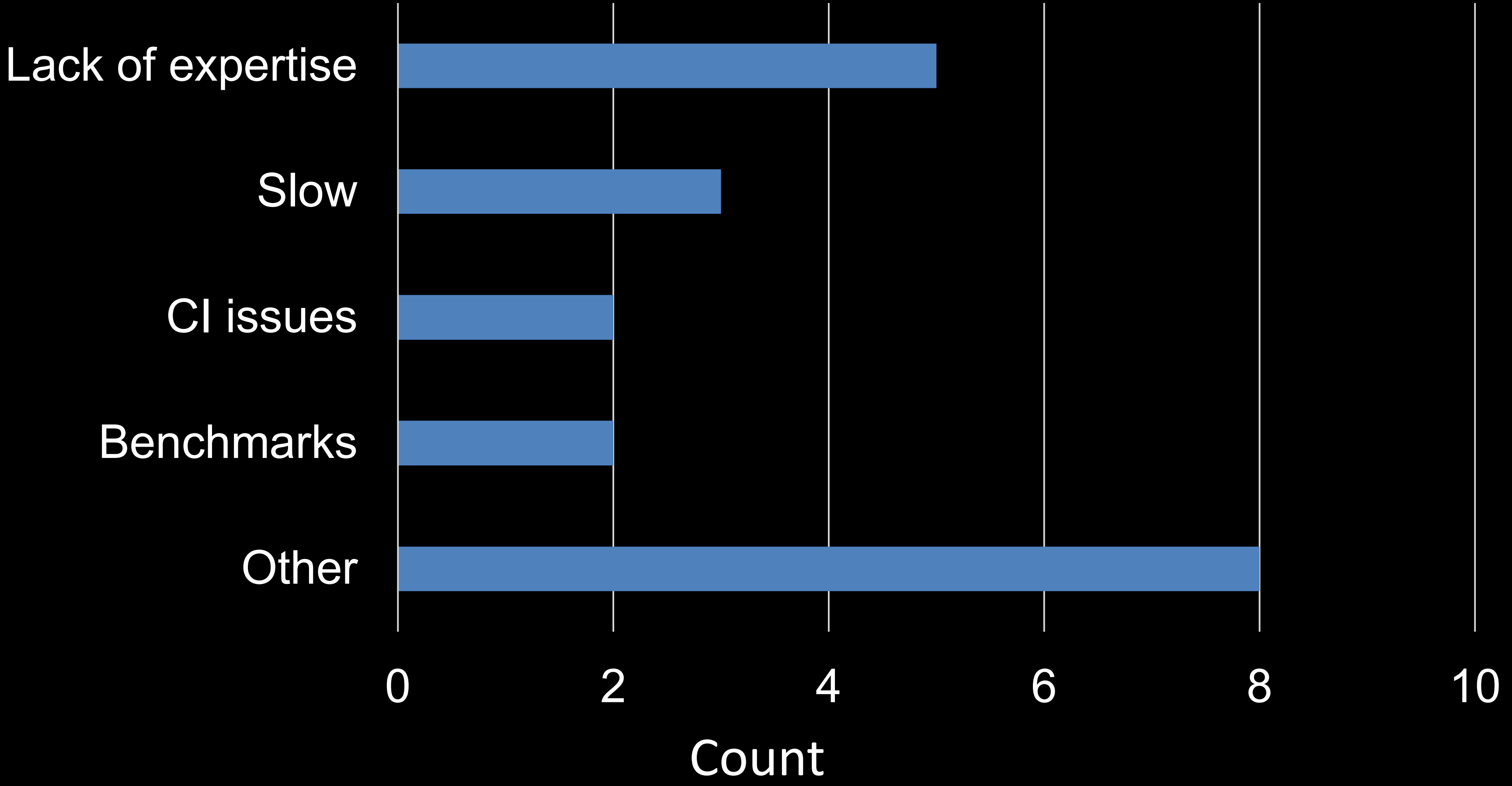
Value Seen in Comm/IT Testing Methods



Challenges to Adapt Comm/IT Methods



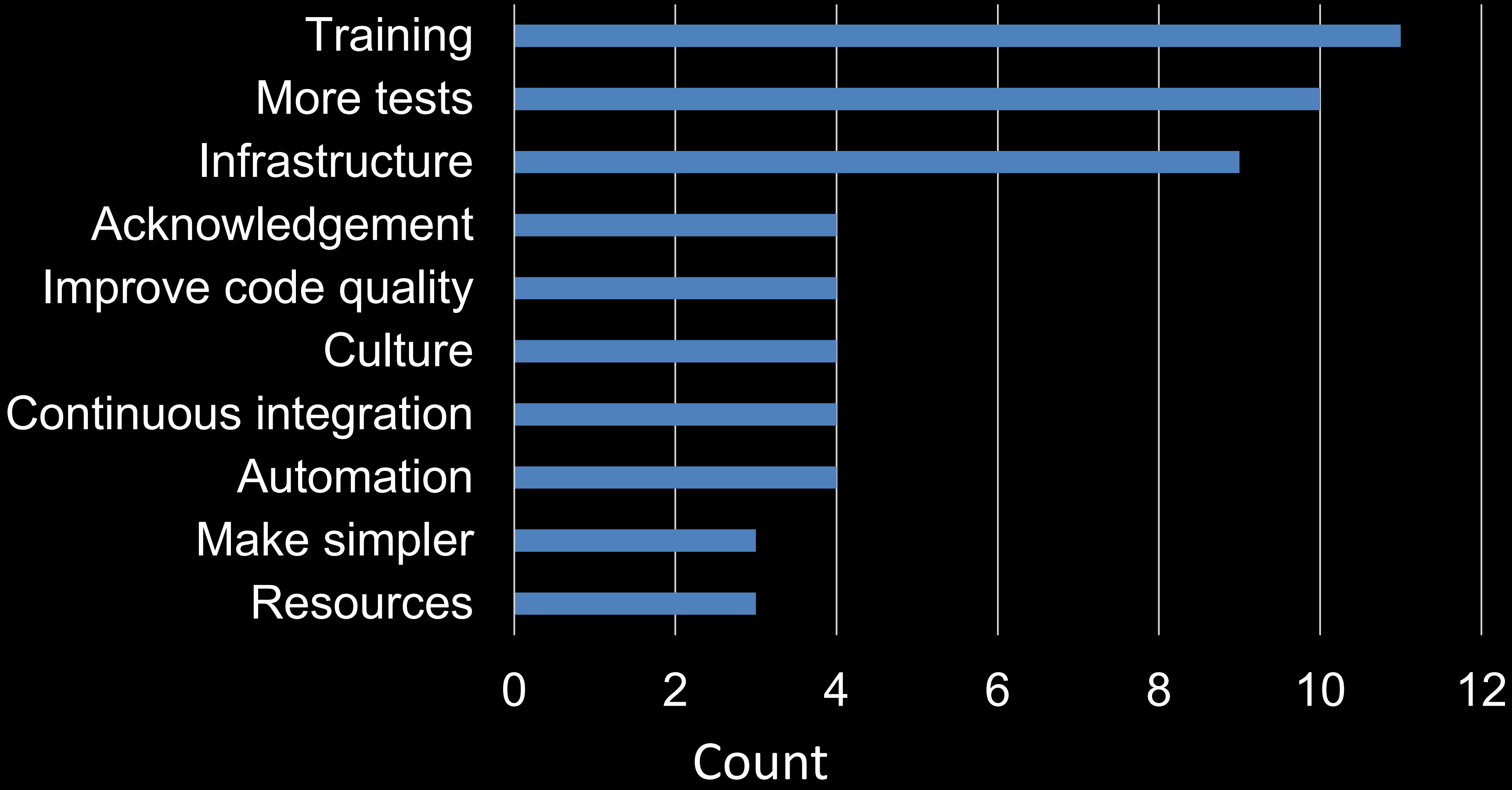
Challenges Not Met by Comm/IT Methods



Part #1 (Testing) Outline

- Demographics
- Level of knowledge research software developers have on testing
- Current testing practices in research software community
- Difficulties to test research software
- Compatibility of Commercial/IT testing techniques
- Improvement of the testing process in research software

Testing Improvements



Discussion

- Researchers pose a clear goal of testing their project
- Complexity associated with the process needs further attention
- Make a culture of testing in the research software community.
- Providing proper training and resources can improve the testing process in research software.

Part #2

Peer Code Review in Research Software

Interview & Survey

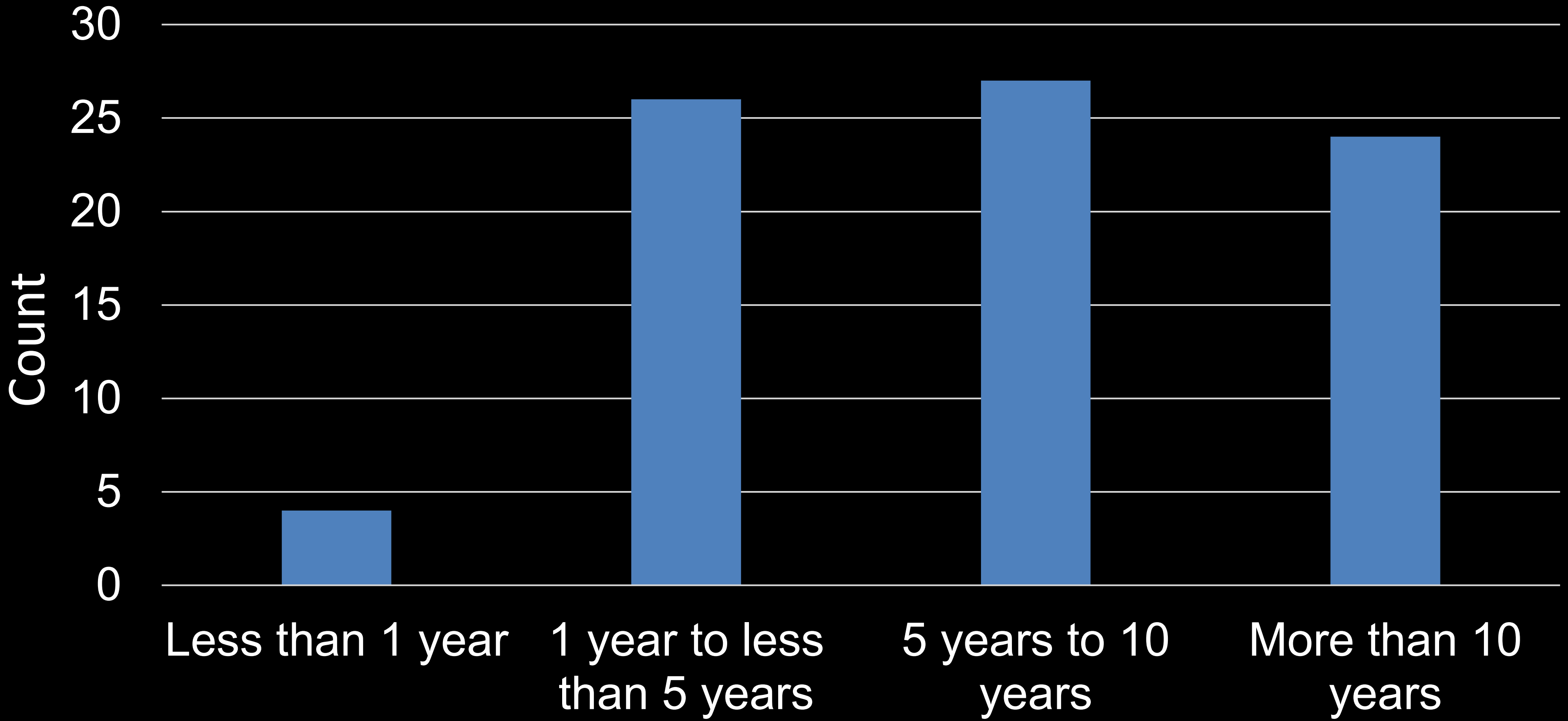
Part #2 (Code Review) Outline

- Demographics
- Current code review practices in research software
- Impacts of the code review process in research software
- Difficulties develpers face during code review
- Potential areas of improvement in the review process

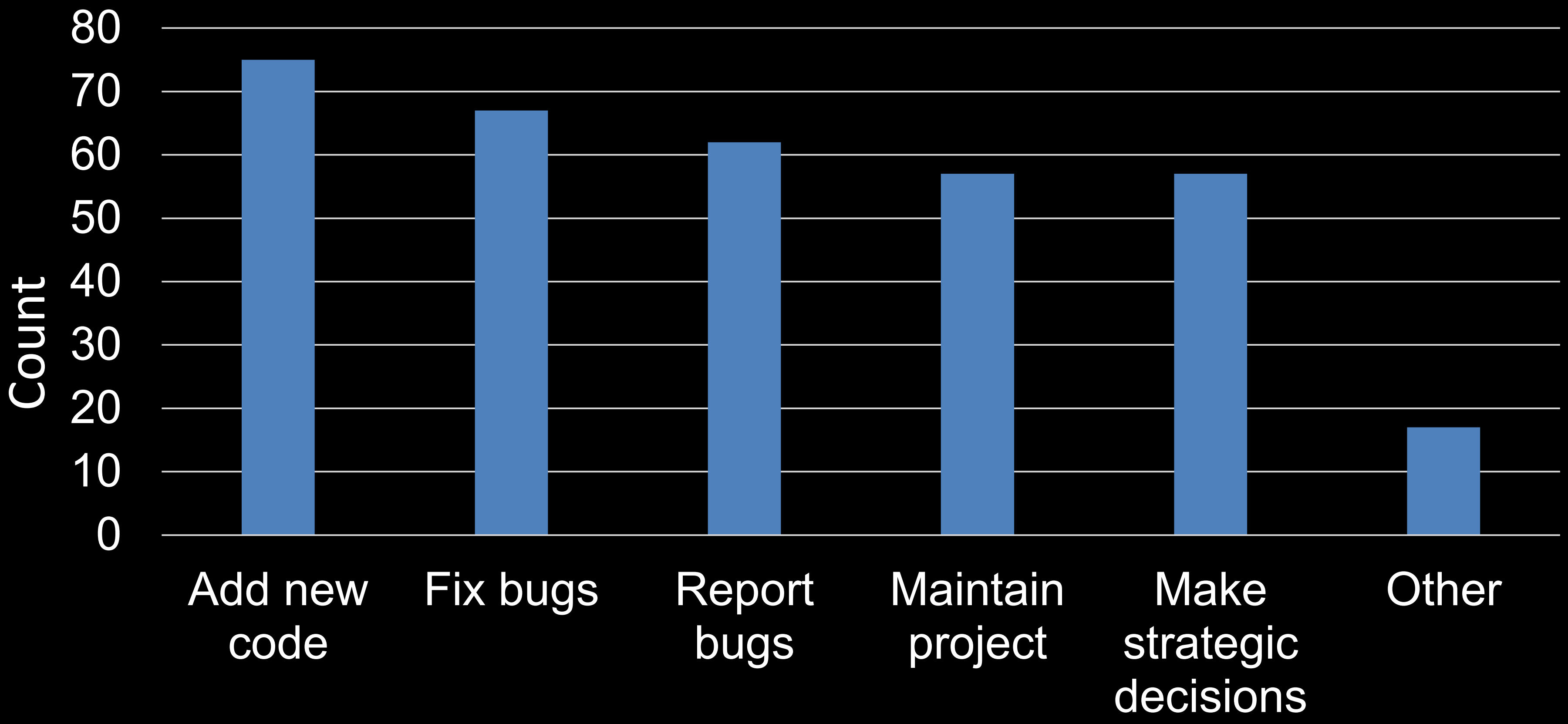
Part #2 (Code Review) Outline

- Demographics
- Current code review practices in research software
- Impacts of the code review process in research software
- Difficulties develpers face during code review
- Potential areas of improvement in the review process

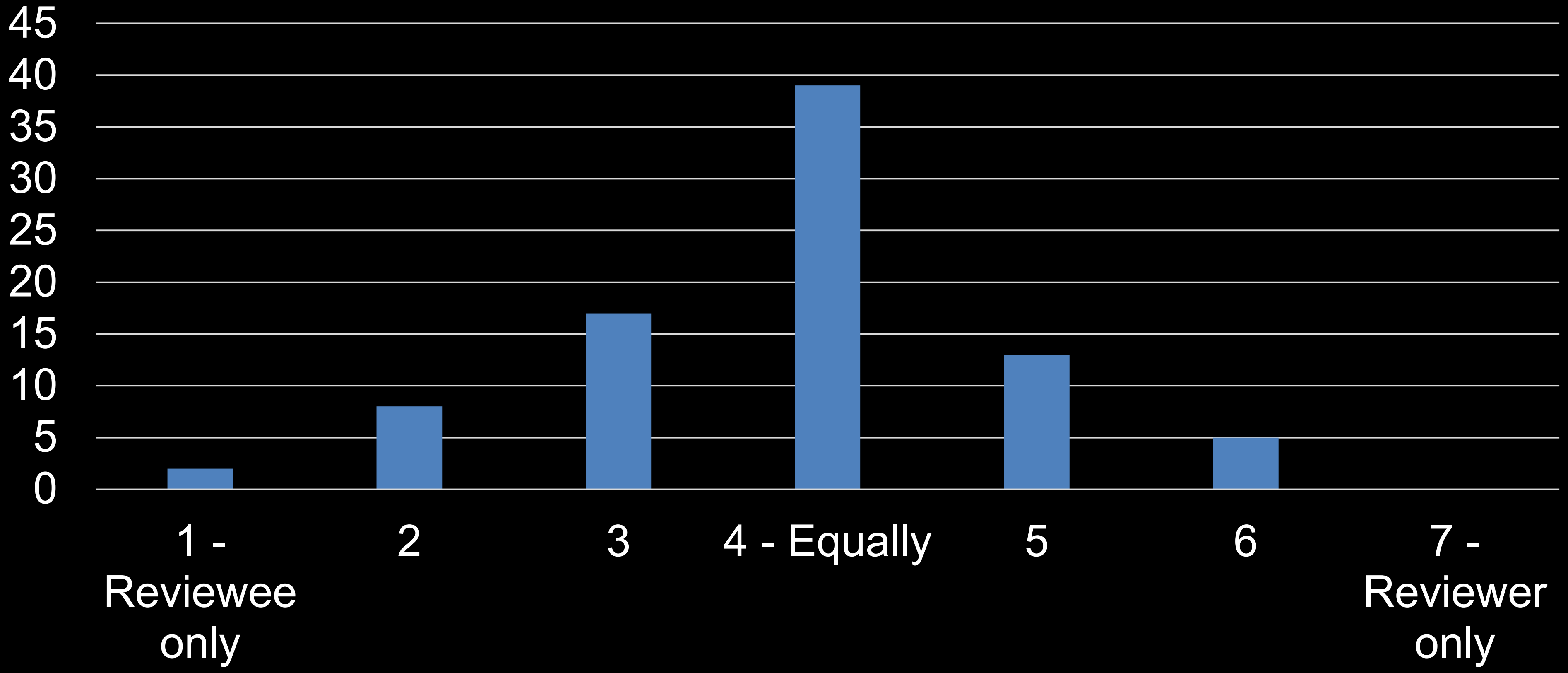
Years Worked



Role


Count
80
70
60
50
40
30
20
10
0
Add new code
Fix bugs
Report bugs
Maintain project
Make strategic decisions
Other

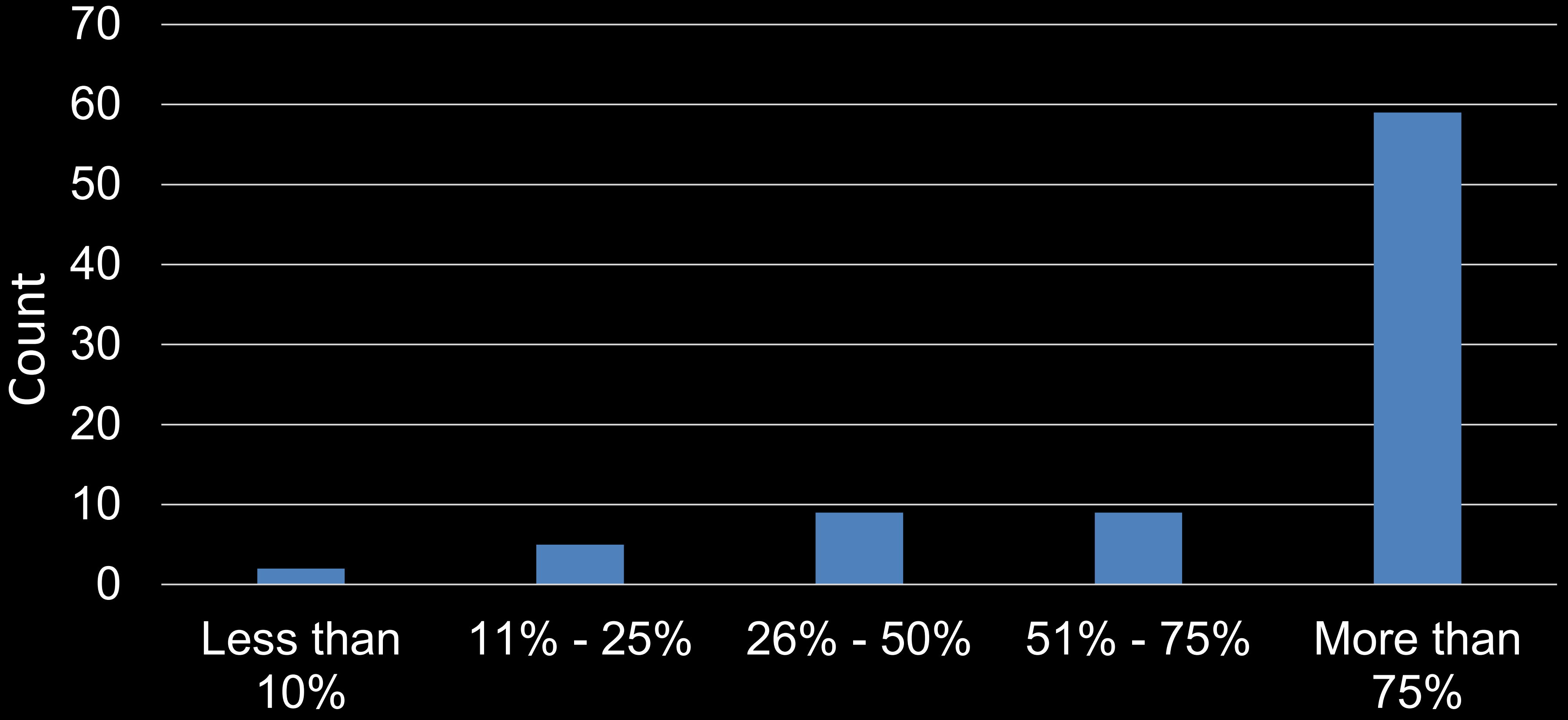
Balance as a Reviewee and Reviewer



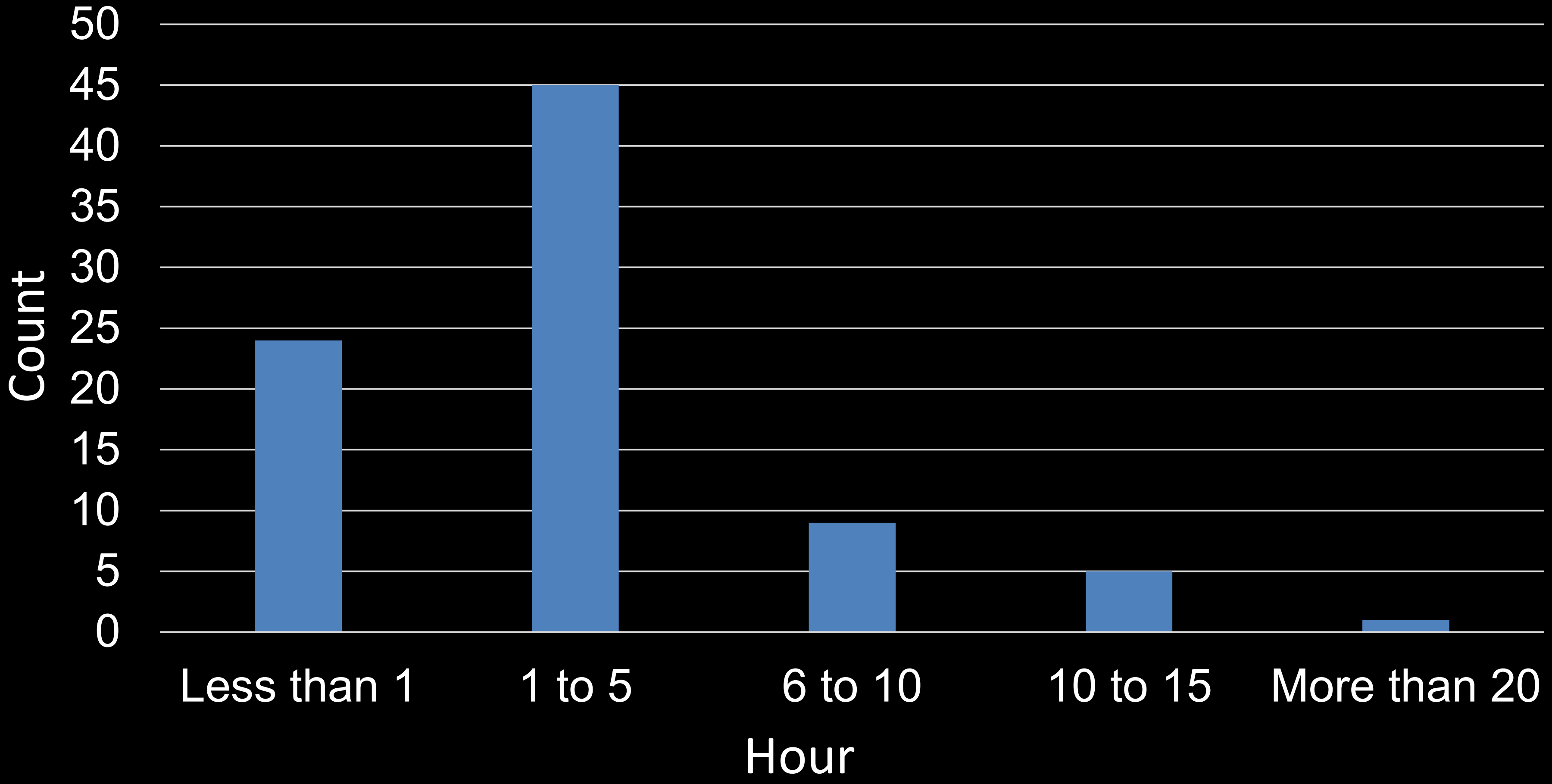
Part #2 (Code Review) Outline

- Demographics
- Current code review practices in research software
- Impacts of the code review process in research software
- Difficulties develpers face during code review
- Potential areas of improvement in the review process

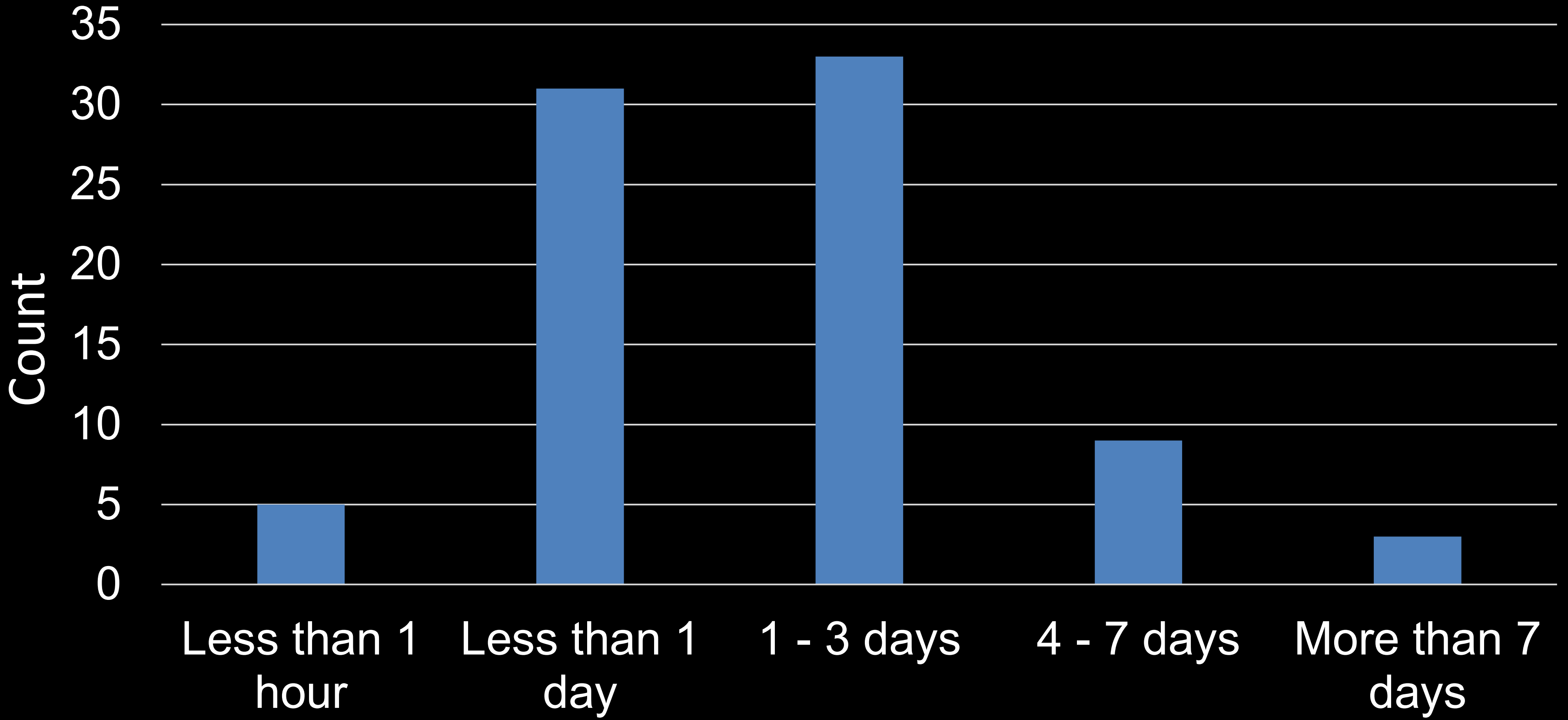
Percentage of Code Undergo Review



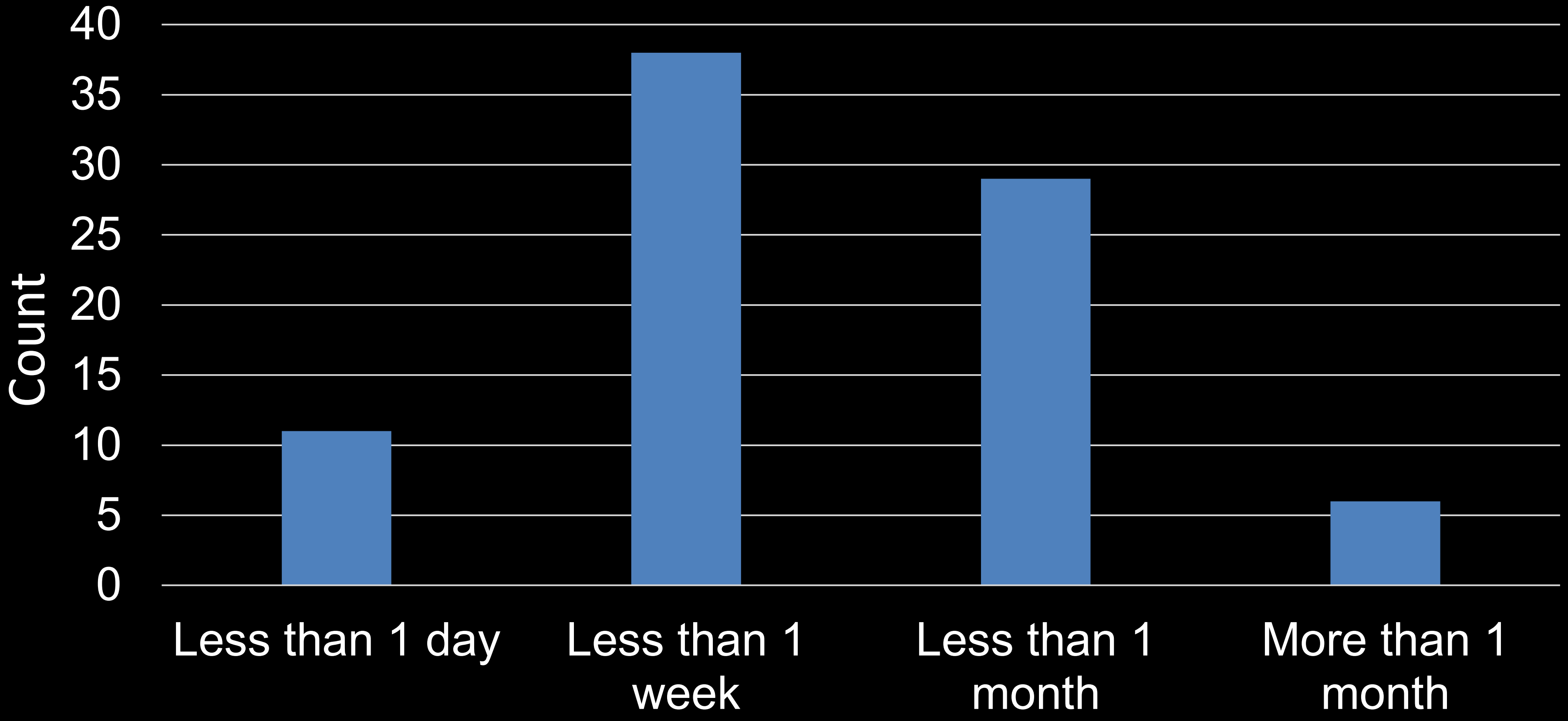
Time Spent on Code review



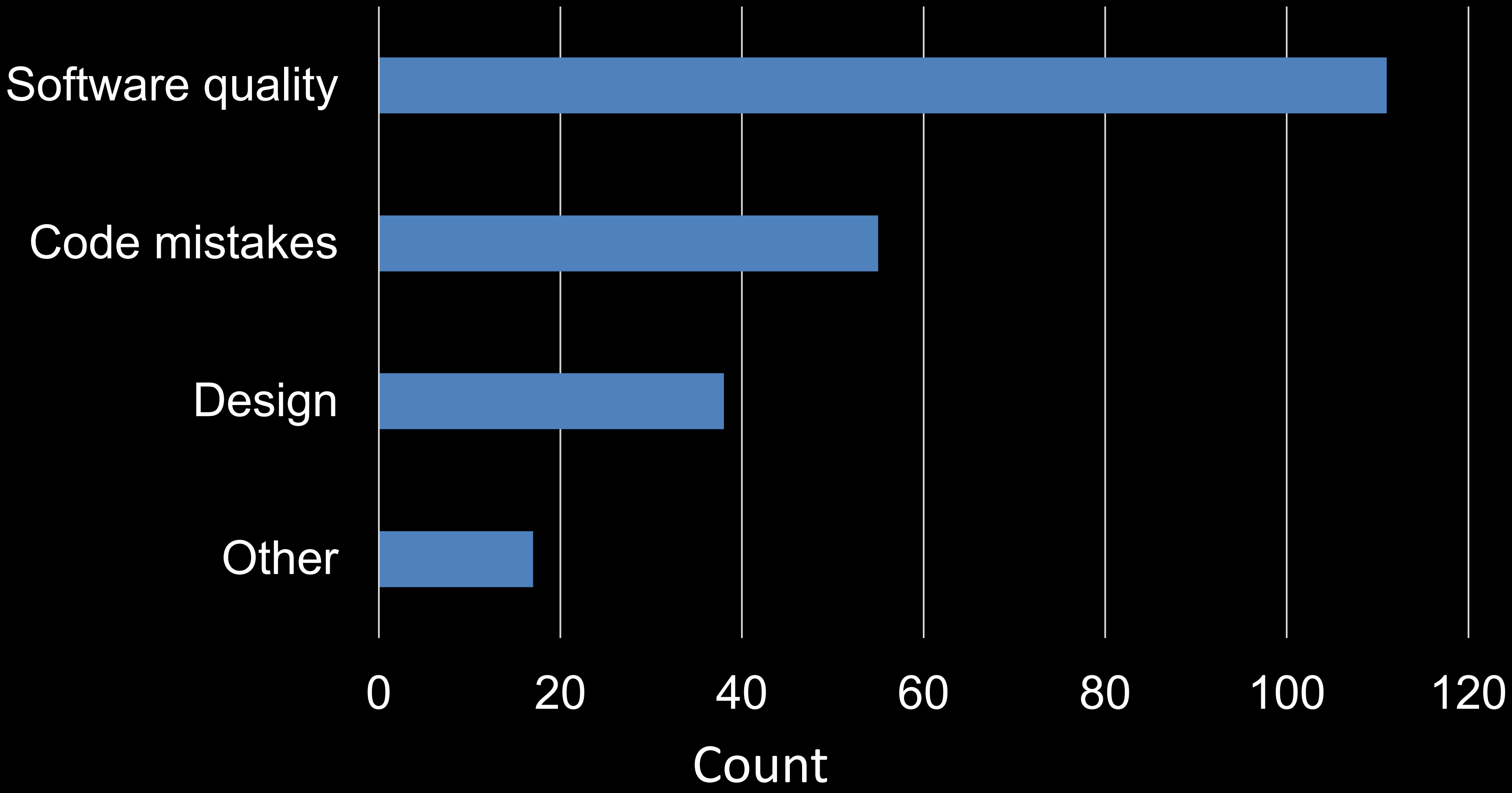
Time For a First Response



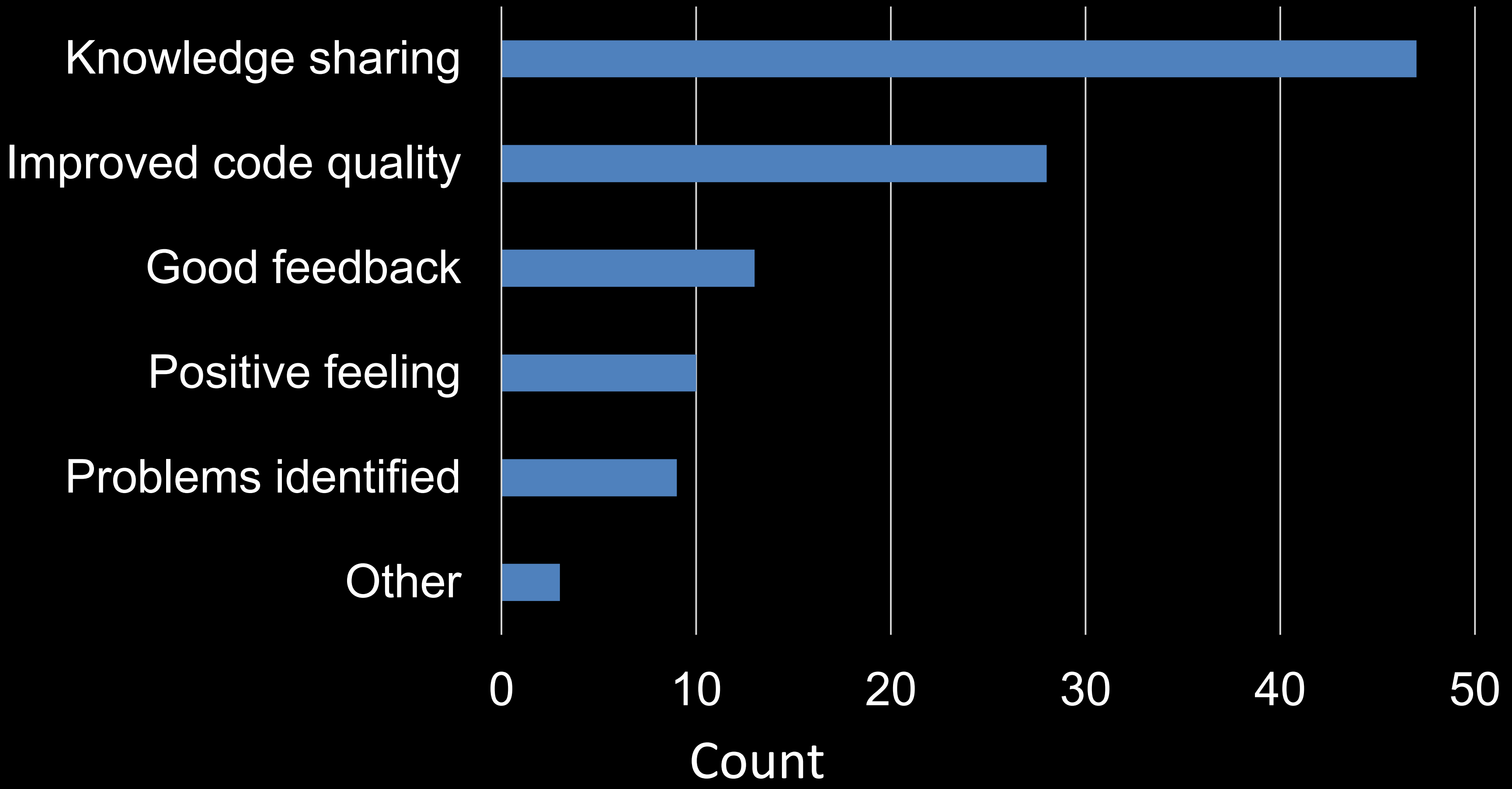
Time For a Final Decision



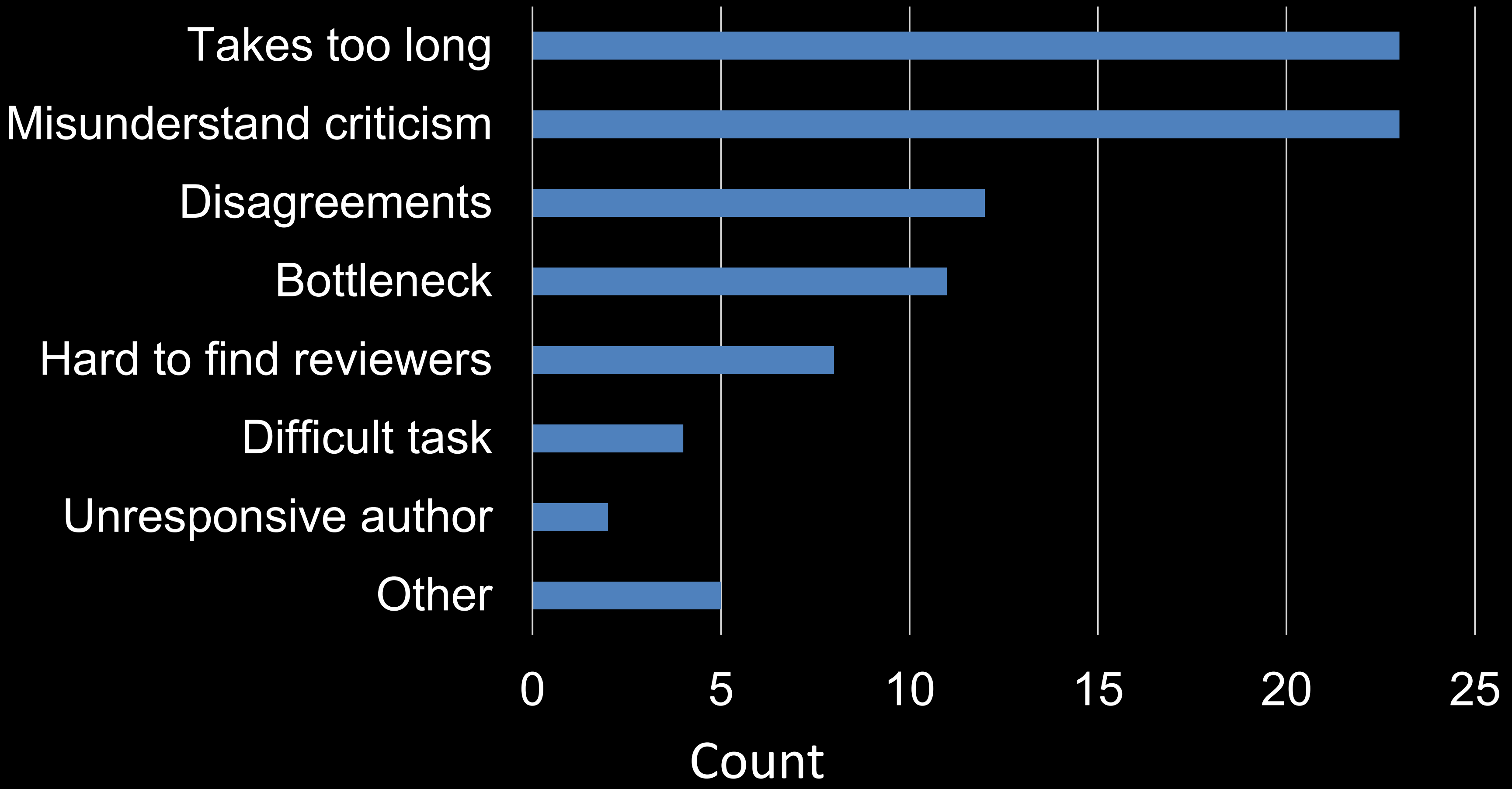
Problems Identified



Positive Experience



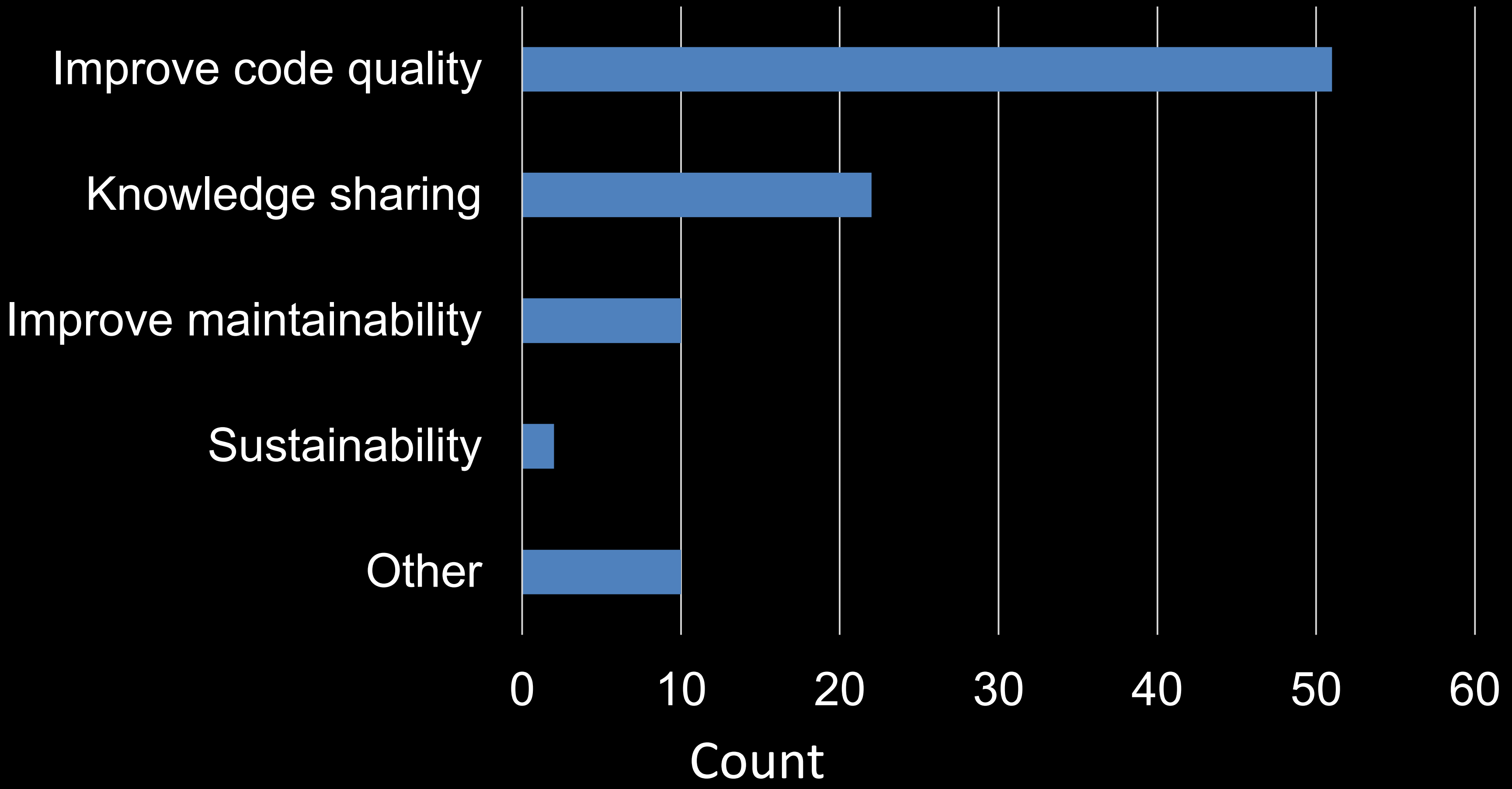
Negative Experience



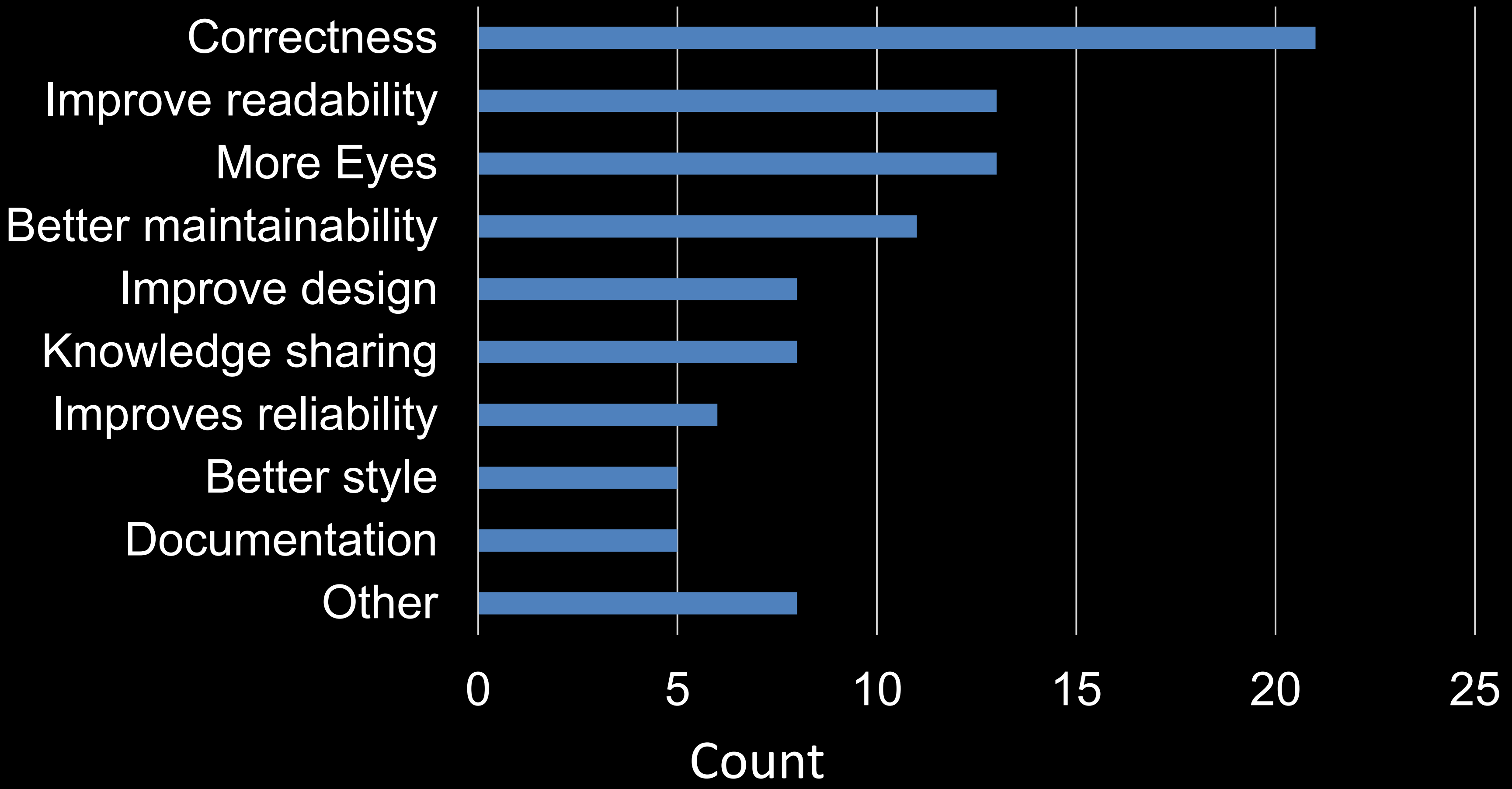
Part #2 (Code Review) Outline

- Demographics
- Current code review practices in research software
- Impacts of the code review process in research software
- Difficulties develpers face during code review
- Potential areas of improvement in the review process

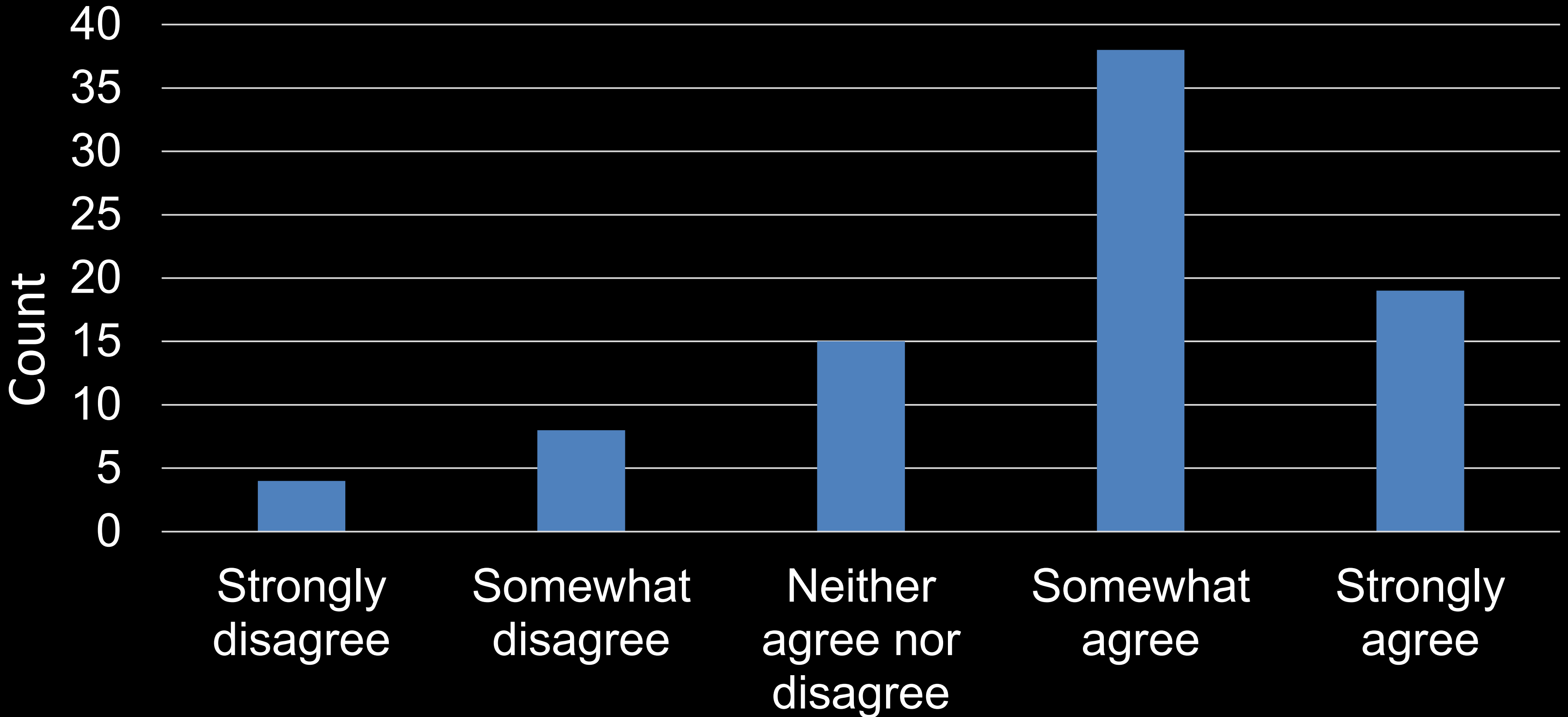
Why Code Review is Important



How Code Review Improves Code



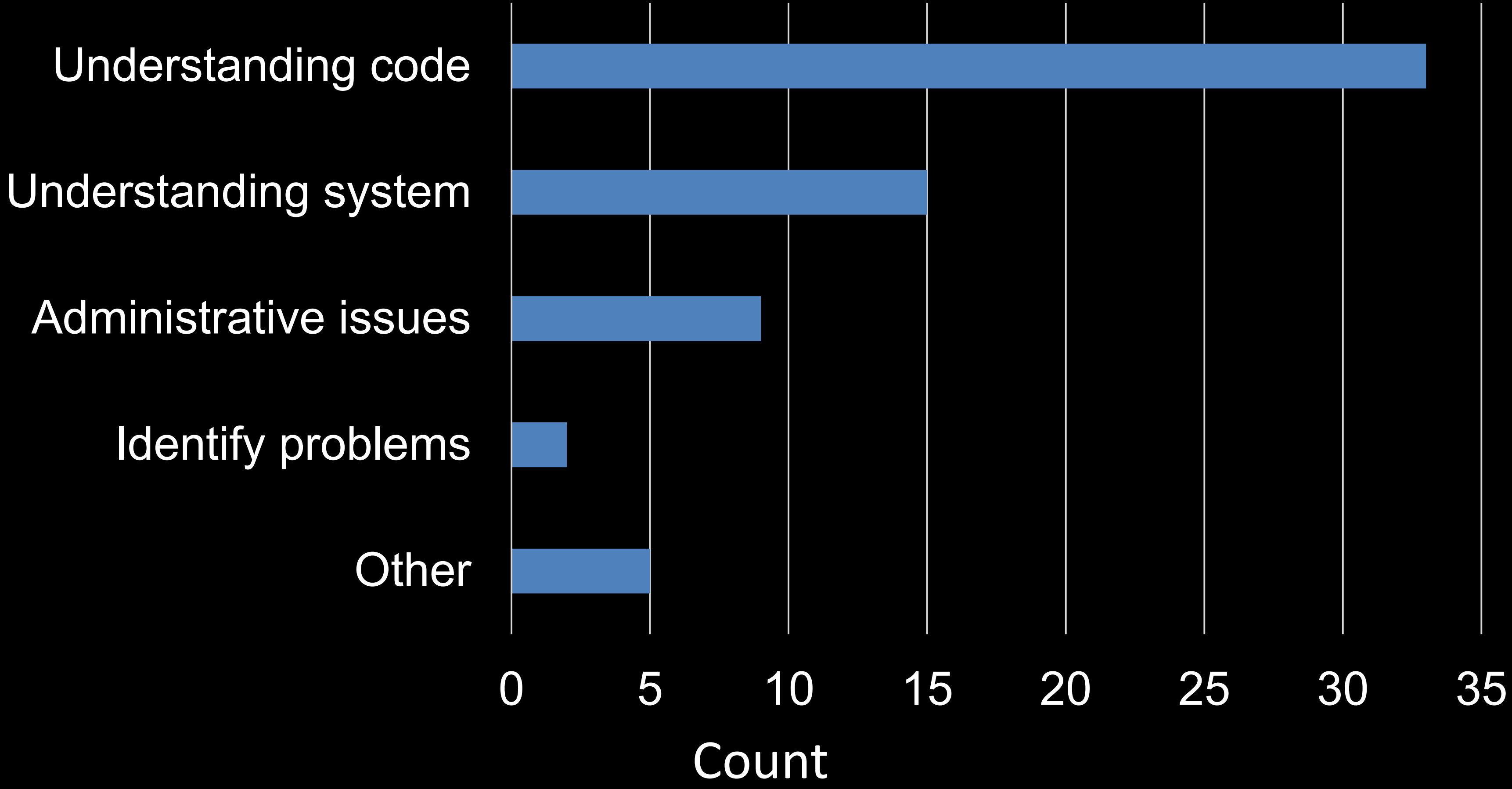
Decrease Code Complexity



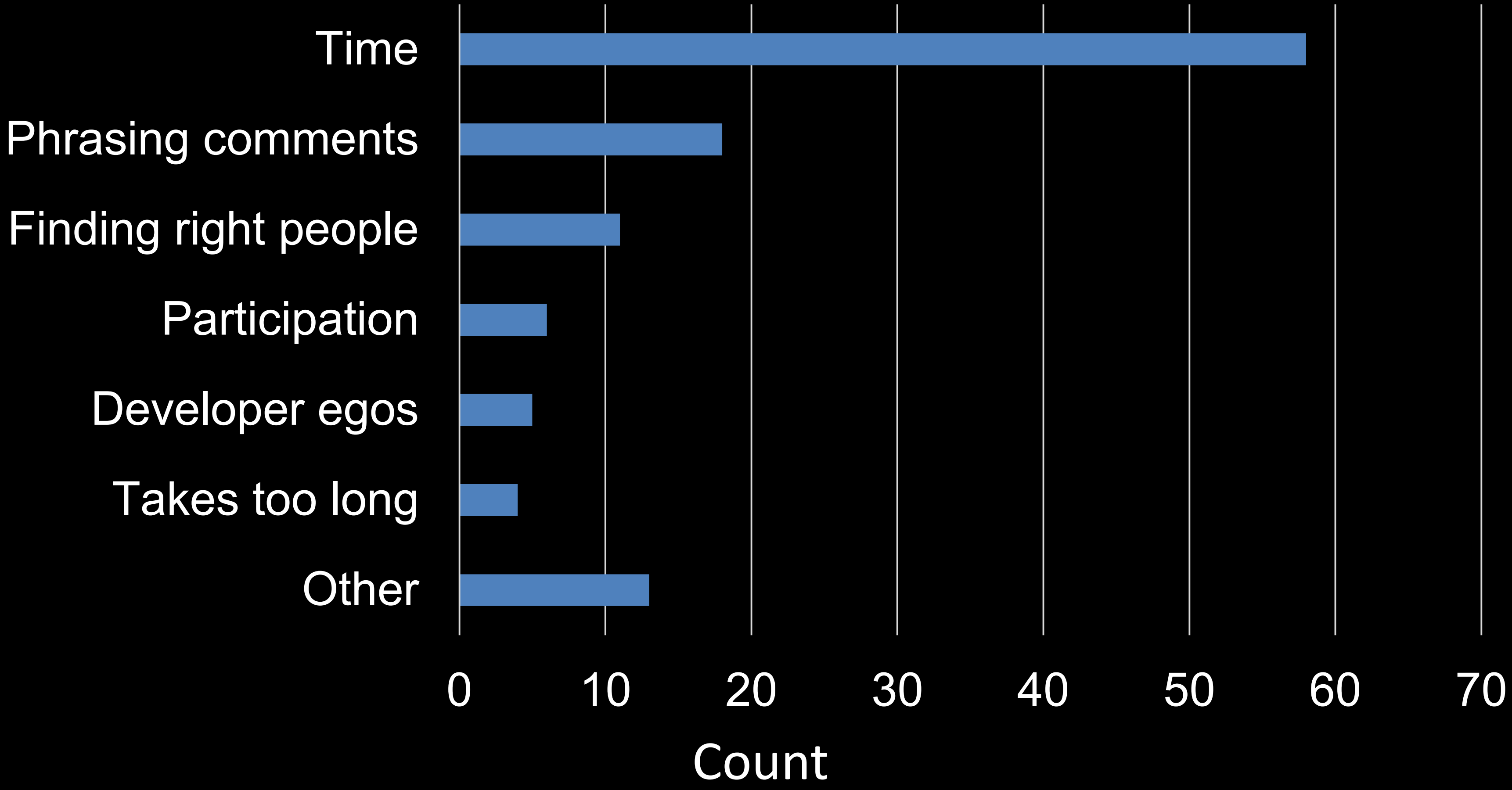
Part #2 (Code Review) Outline

- Demographics
- Current code review practices in research software
- Impacts of the code review process in research software
- Difficulties develpers face during code review
- Potential areas of improvement in the review process

Challenges


Understanding code
Understanding system
Administrative issues
Identify problems
Other
0
5
10
15
20
25
30
35
Count

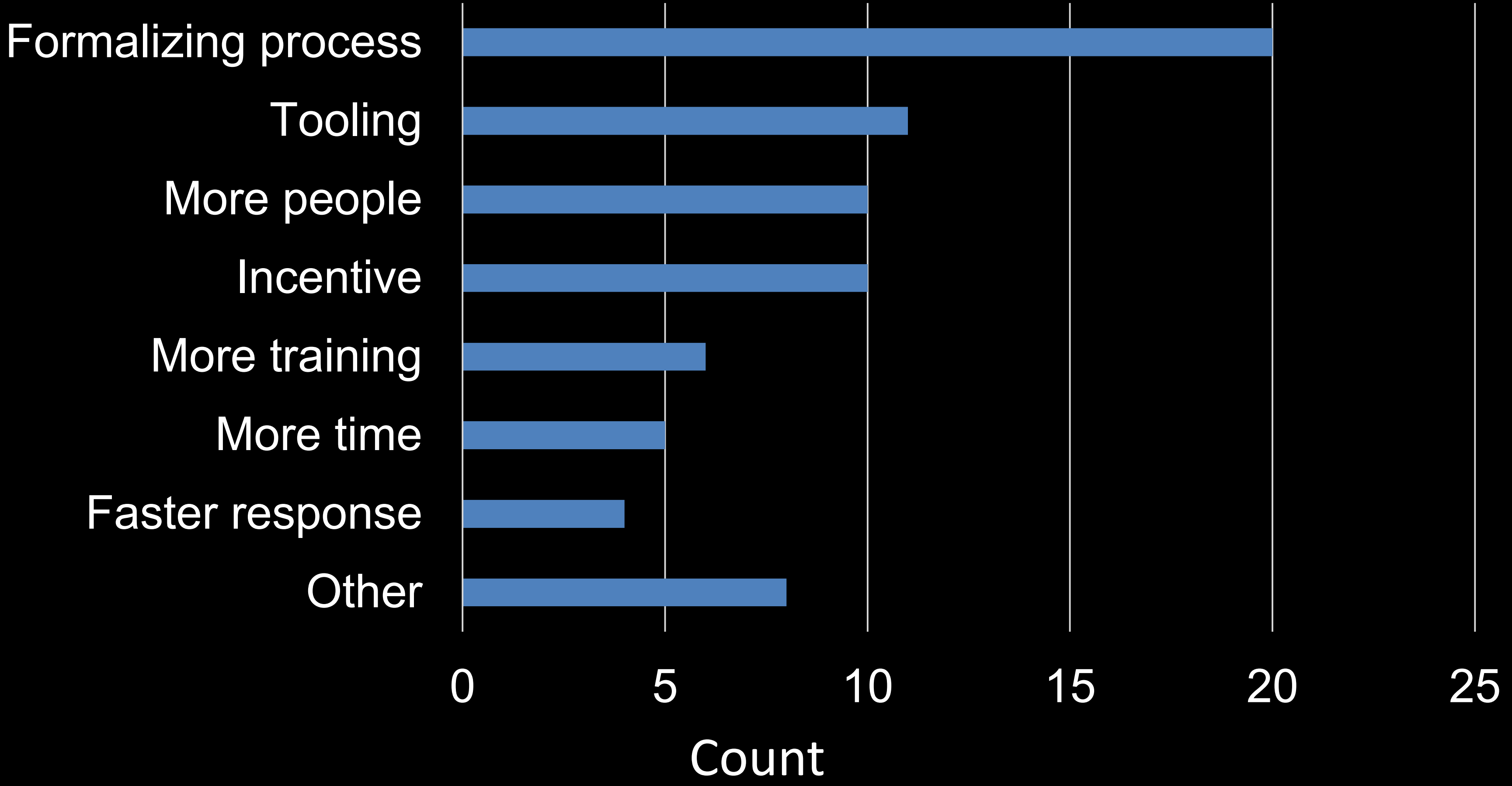
Barriers


Time
Phrasing comments
Finding right people
Participation
Developer egos
Takes too long
Other
0
10
20
30
40
50
60
70
Count

Part #2 (Code Review) Outline

- Demographics
- Current code review practices in research software
- Impacts of the code review process in research software
- Difficulties developers face during code review
- Potential areas of improvement in the review process

Improvements



Discussion

- Research software developers employ an informal code review process
- Code review has an overall positive impact
- Most common difficulty reported by participants is finding time to to do it and understand other people’s code.
- Formalizing the review process by including more people, more training, and providing compensation could potentially improve the code review process.

Recommendations - Testing

Recommendations - Testing

- Provide enough training on software testing to all kinds of research software developers ranging from graduate students to experienced researchers

- Incorporate more tests that can solve specific needs of the research software

- Provide infrastructure support, for example, a public service for testing including many-tier pricing structure for machine time and a sophisticated testing dashboard

Recommendations - Testing

- Provide automation for setting tests and analysis of the results
- Improve continuous integration system to facilitate a better way of testing, especially, the incoming tests during down time
- Make a culture of testing in the team and encourage others by sharing the benefits from the experience of testing
- Improve the quality of the code so that developers can write tests easily

Recommendations - Testing

- Provide proper acknowledgement of developers for contributions in testing

- Make the testing process simpler so that it is easy to adopt in the project

- Provide enough resources to developers so that they can utilize the resources to develop test suits

Recommendations – Code Review

Recommendations – Code Review

- Make code review process more formal with a structured guideline for each step of the process

- Try to ensure at least one science review and one technical review

- Include automatic tools in the code review process and train your peer reviewers the best practices to use the tool

Recommendations – Code Review

- Encourae more people to participate in the review process and allocate some time to do the review
- Provide incentives or rewards to reviewers to participate in code review
- Allocate sufficient time in the development process to perform code review
- Provide faster feedback to any incoming review request

Recommendations – Code Review

- Train reviewers on how to phrase good feedback

- Train developers to forget their egos and accept comments from the reviewers to improve their code

- Make the overall code review process faster

- Provide necessary support from the administrative level that encourages people to participate in the code review process

Acknowledgement

Dr. Jeffrey Carver
University of Alabama

Dr. Hai Ah Nam
Los Alamos National Laboratory

Dr. Roland Haas
National Center for Supercomputing Applications

Dr. George Thiruvathukal
Loyola University Chicago

Dr. Danny Perez
Los Alamos National Laboratory

Dr. Gabrielle Allen
National Center for Supercomputing Applications

Dr. David Bernholdt
Oak Ridge National Laboratory

Dr. J. Dave Moulton
Los Alamos National Laboratory

Dr. Daniel Katz
National Center for Supercomputing Application

NSF grant 1445344

Nasir Eisty
neisty@calpoly.edu