AN EMPIRICAL STUDY OF PRACTITIONERS' PERSPECTIVES ON GREEN SOFTWARE ENGINEERING

- Empirical studies of how developer decisions impact energy consumption (design patterns, web servers, refactoring, code obfuscation)
- Energy-optimization decision framework (SEEDS)
- Energy-directed test suite optimization

MOTIVATION

- Increased use of smart phones and data centers has lead to increased concerns about the amount of energy they consume

- Energy usage concerns have lead to increased interested from the research community

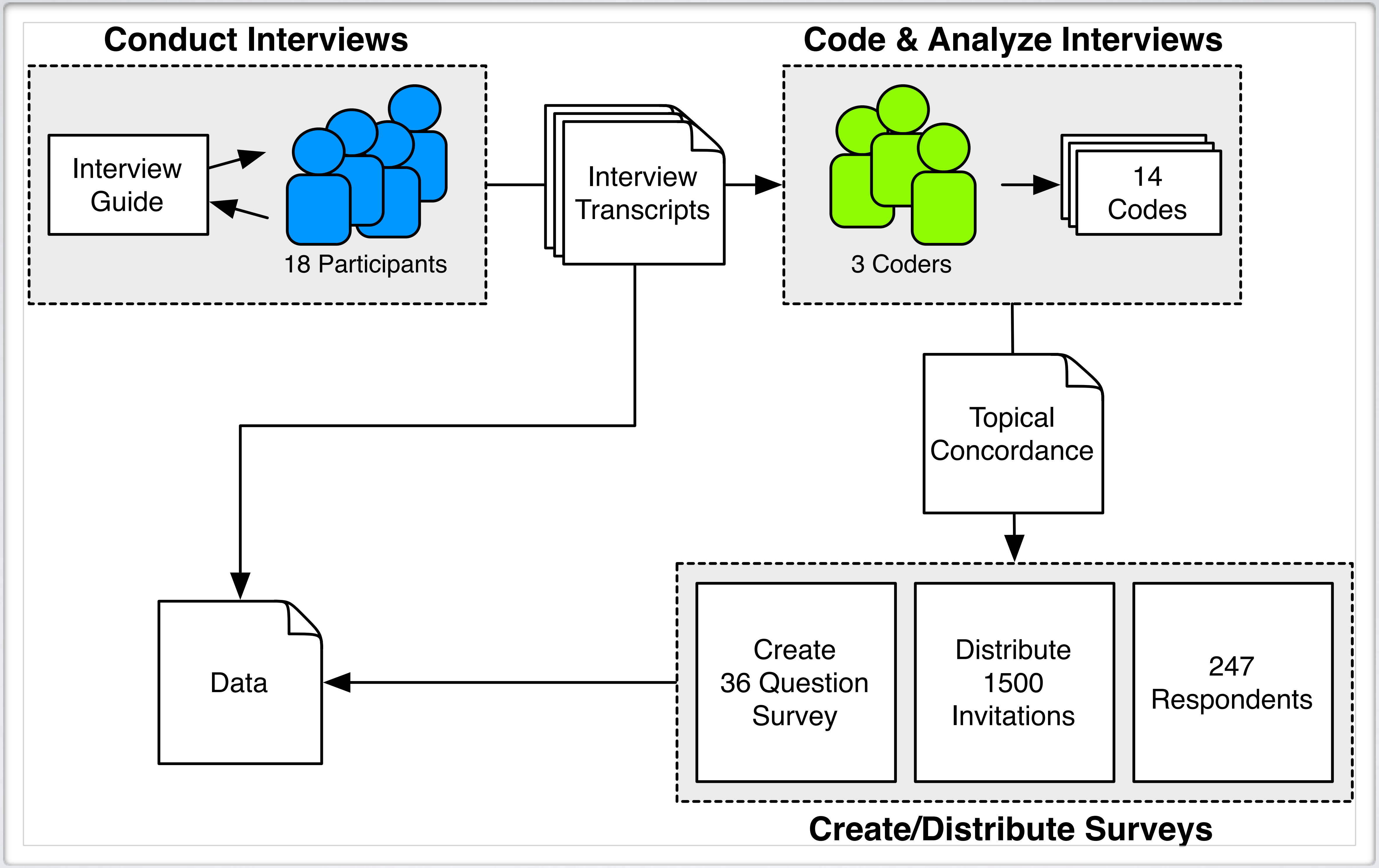
- Is the research community targeting problems that are important to practitioners?

APPROACH

Conduct an empirical study of practitioners' perspectives on green software engineering—how they think about energy when they write requirements, design, construct, test, and maintain their software.

The results of the study can motivate and guide green software engineering research

METHODOLOGY



Interview and surveys are complimentary

INTERVIEWS

- Goal: organically learn about how interviewees think about energy during software development
- Protocol: Semi-structured; at interviewee's office; 30–60 minutes; audio recorded; two interviewers
- Participants: Purposive sampling of Microsoft employees with interests in energy; expanded via the snowball process
- Topical concordance produced by open, axial, and selective coding of the transcripts

SURVEYS

- Goal: quantitatively assess the information learned from the interviews
- Protocol: 15 minute survey composed of Likert response statements (36 total) drawn from the topical concordance
- Participants: Microsoft employees based on their position in the organizational chart (1500 invitations, 247 responses, 16%)

FINDINGS

- In what domains is energy usage of concern?
  - Considered all respondents
- What are experienced practitioners' perspectives (grouped by SWEBOK)?
  - Considered respondents whose projects Sometimes, Often, or Almost Always have energy usage requirements or goals (121 of 247, 49%)

WHERE IS ENERGY USAGE A CONCERN?

Respondents were grouped by the respondent's most closely related project.

Data Center
(Azure, Bing)

"In the data center, any watt that we can save is a watt we don't have to pay for."

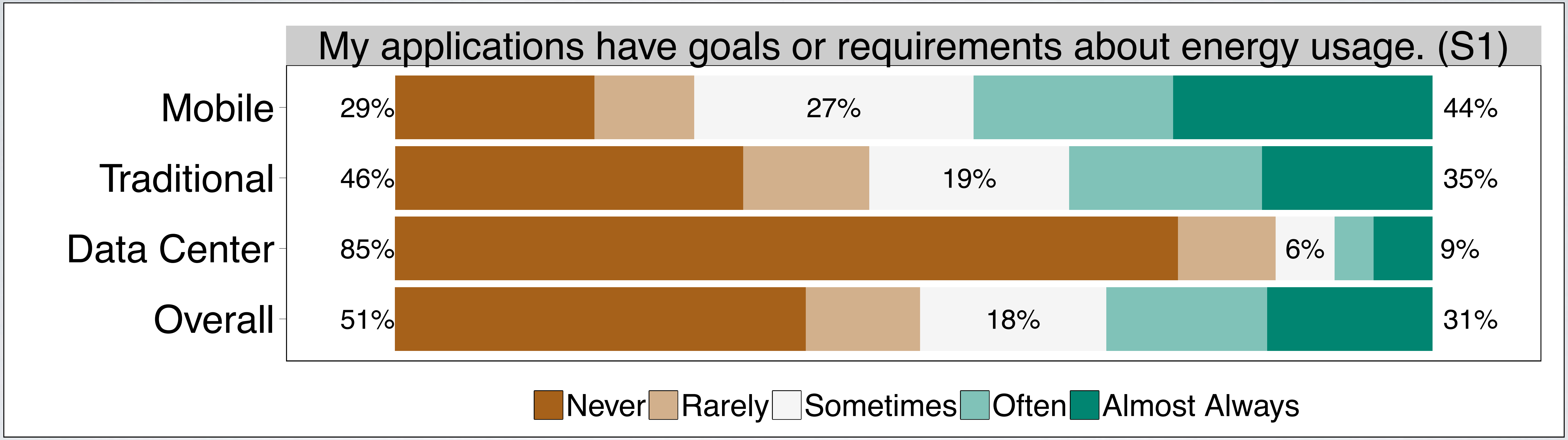
Mobile
(Surface, Windows Phone)

"Battery life is very important in mobile apps and I would like to see improvement there."

Traditional
(Windows, XBox, Skype)

Initial assumption was that energy usage requirements are more common for mobile and data center projects than traditional projects.

WHERE IS ENERGY USAGE A CONCERN?



- 76% of Traditional developers have extensive experience with Mobile.
- 26% of Data Center developers have extensive experience with Mobile.

"We want to minimize energy, but for developers performance is more critical than saving energy."

"The primary goal is the user experience, not saving money."

PERSPECTIVES ON REQUIREMENTS

- What do typical requirements look like?
- How often do practitioners make tradeoff between other requirements and energy usage?

Energy-usage requirements are more often desires rather than specific targets.

Desires

"No specific goals for energy usage, just 'don't be bad'."

"Considerations on backgrounds tasks as well as those that use the radio are always in the back of my mind."

"The goal is to accomplish something without making the user annoyed about battery drain."

Specific Targets

"Performing [user scenario] should not use more than X mA."

"Under normal usage, a device with an X Wh battery should last for Y hours."

"Turn-by-turn guided navigation should not drain more battery than a car can charge."

Energy-usage requirements are often stated in terms other than energy usage.

Traditional Metrics:

"I consider running time and that 'seems' to suggest battery life."

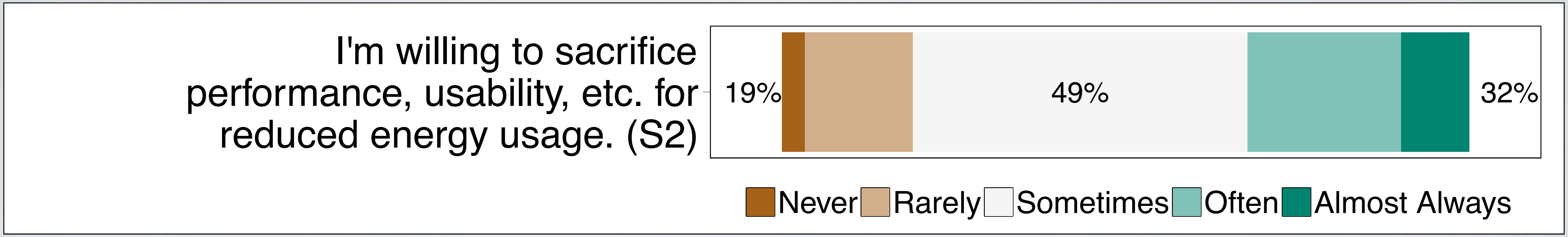
"Most people think that power savings = CPU reduction. This is somewhat true in a broad sense, but is only a small part of the picture. The problem is that it's easy to measure CPU utilization (and hence reduction), but it's very hard to translate any of this to actual power savings. Many people have spent a lot of time that ultimately had no benefit."

Energy-usage requirements focus on idle time.

“We’re trying to prioritize idle battery consumption down to zero. Being active is going to drain the battery. But the thing that’s going to [annoy] people, is if I wan’t using it and my battery is dead.”

“I haven’t thought about that, actually, when an app is in the foreground and we’re trying to still save battery.”

Experienced practitioners are often willing to sacrifice other requirements for reduced energy usage.



81% answered Sometimes, Often, or Almost Always.

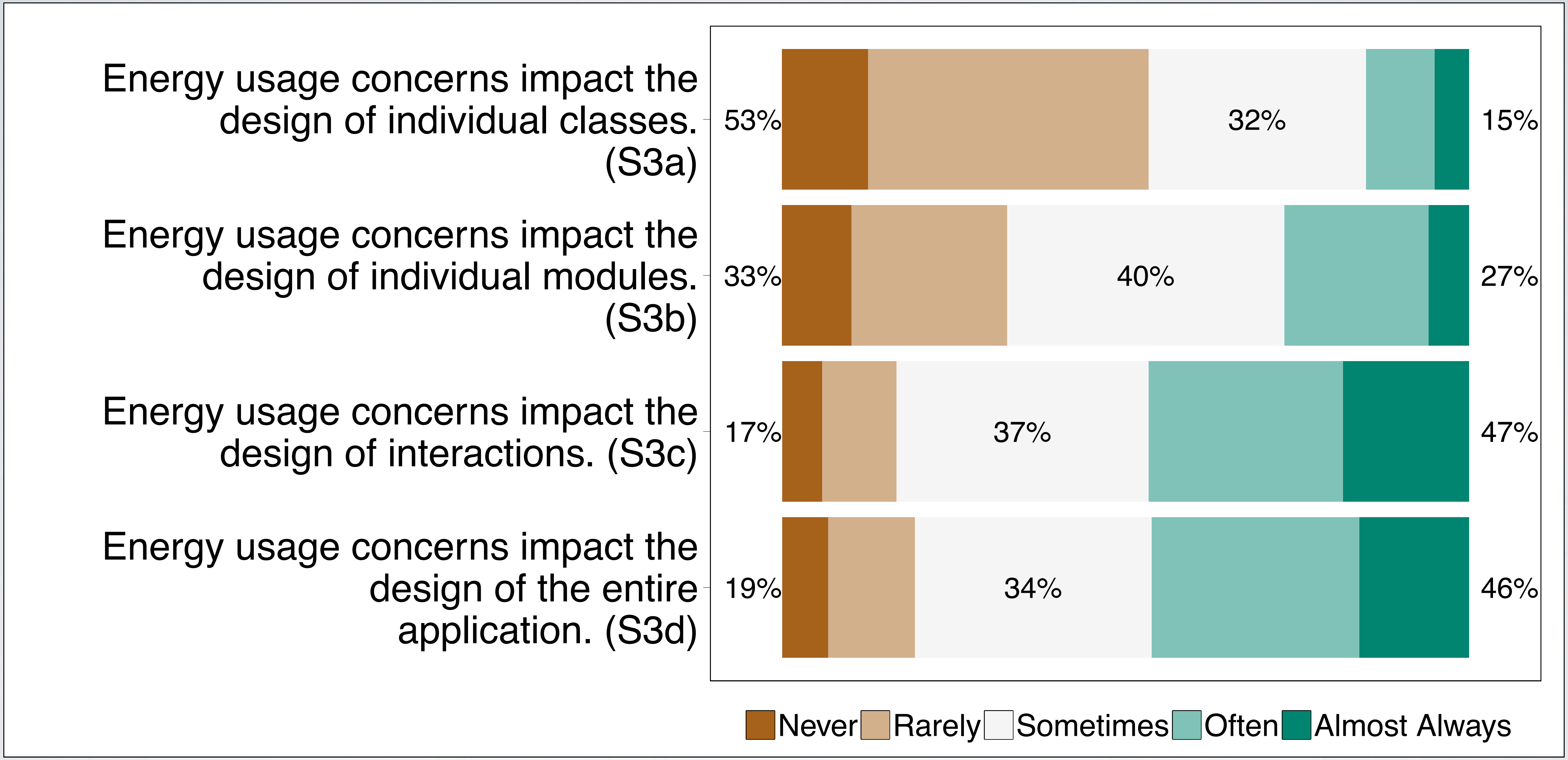
"The entire experience [designing the user interface] was a series of compromises between what the designers wanted and battery concerns."

"There is always a tradeoff between battery life vs. performance/feature."

PERSPECTIVES ON DESIGN

- How do energy concerns impact different aspects of the design process?
- What contexts do practitioners consider?
- Do general (anti-)patterns that lead to good or bad energy usage exist?

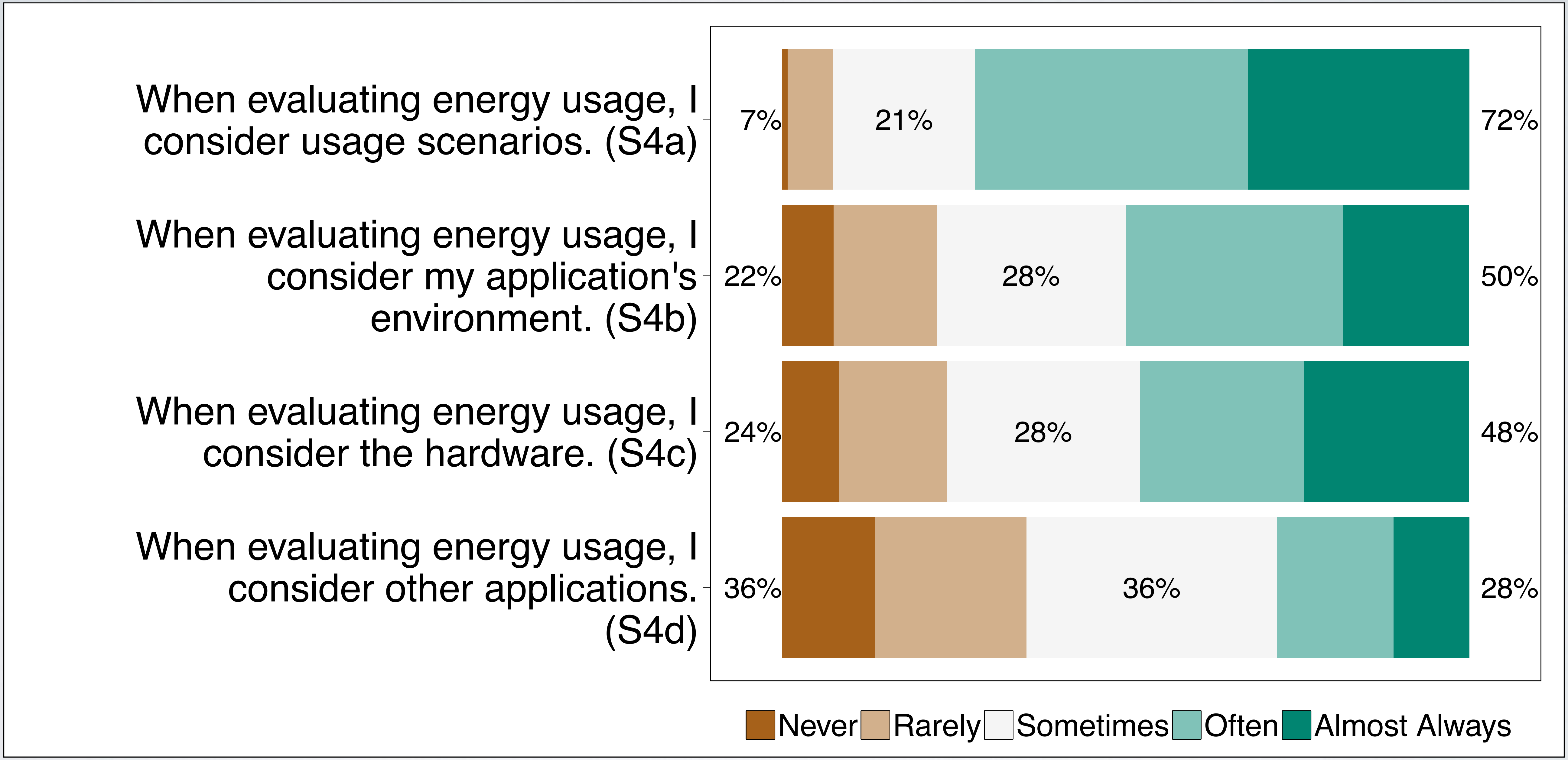
Concerns about energy usage impact how applications are designed



More than 50% indicated that design of modules, interactions, and applications are impacted Sometimes, Often, or Almost Always.

"It's not a bug fix to get power efficiency, it's a design change."

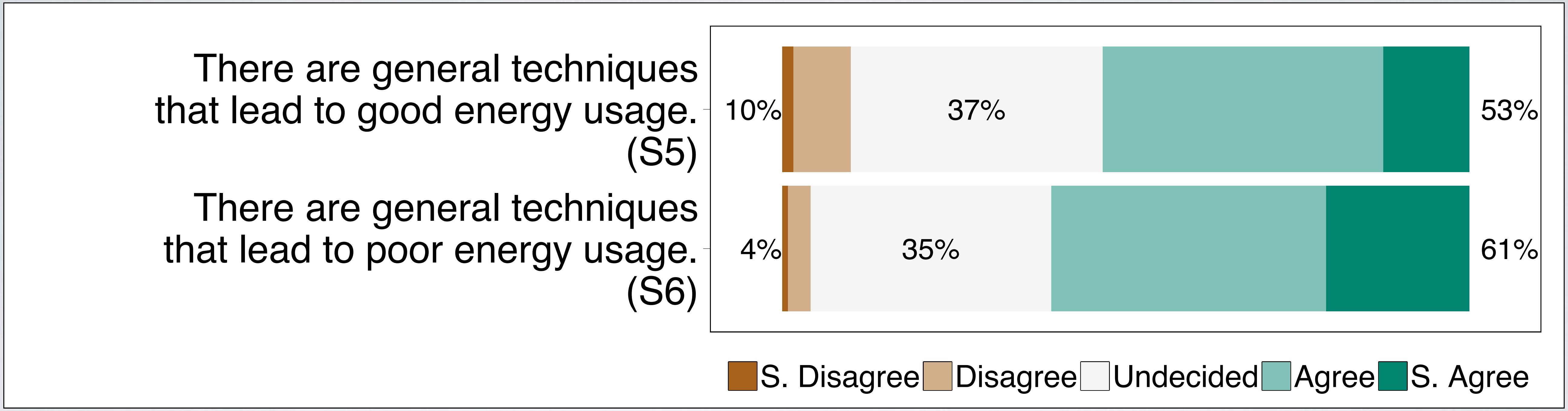
When evaluating energy usage, practitioners consider usage scenarios most often and other applications least often.



- 93% consider user scenarios Sometimes, Often, or Almost Always.
- 64% consider other applications Sometimes, Often, or Almost Always.

"We've started to look at telemetry to figure out realistic goals."

Practitioners believe there are techniques that lead to good and bad energy usage

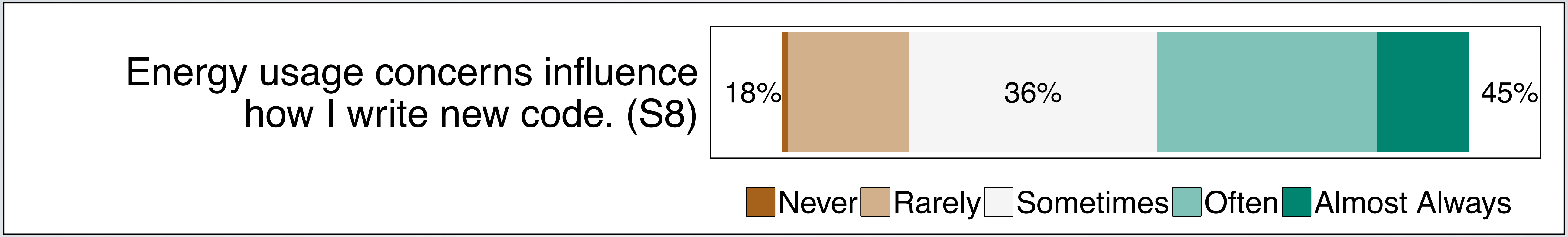


- In general, the lists of good and bad techniques are inverses.
- However, there are cases where the lists contradict each other:
  - “offloading computation to the cloud” reduces energy usage
  - “decreasing radio usage” increases battery life
- The most commonly mentioned good techniques are: event-based architectures, coalescing timers, and efficient algorithms.

PERSPECTIVES ON CONSTRUCTION

- Do energy concerns influence how new code is written?
- Do practitioners believe that they have accurate intuitions about energy usage?
- How would practitioners like to learn how to improve energy usage?
- Who should be responsible for energy usage?

Energy concerns influence how developers write new code

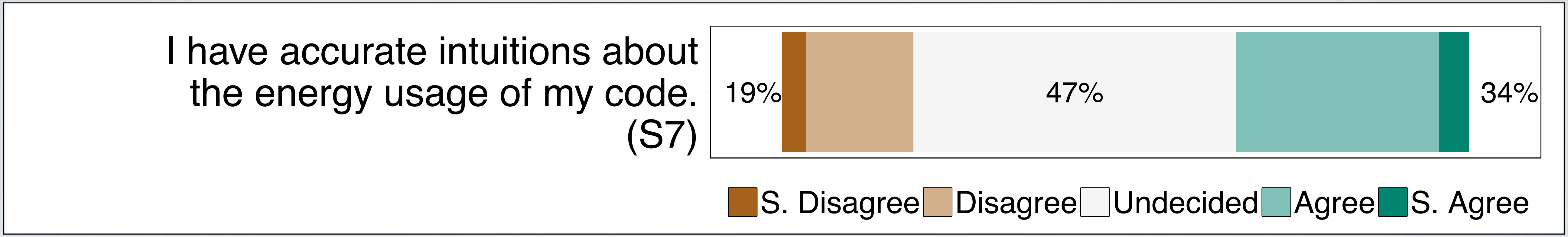


83% consider energy concerns Sometimes, Often, or Almost Always.

This is contrary to what we expected:

"Only when meeting performance goals becomes egregious in terms of power, then we negotiate a compromise that balances performance and power consumption."

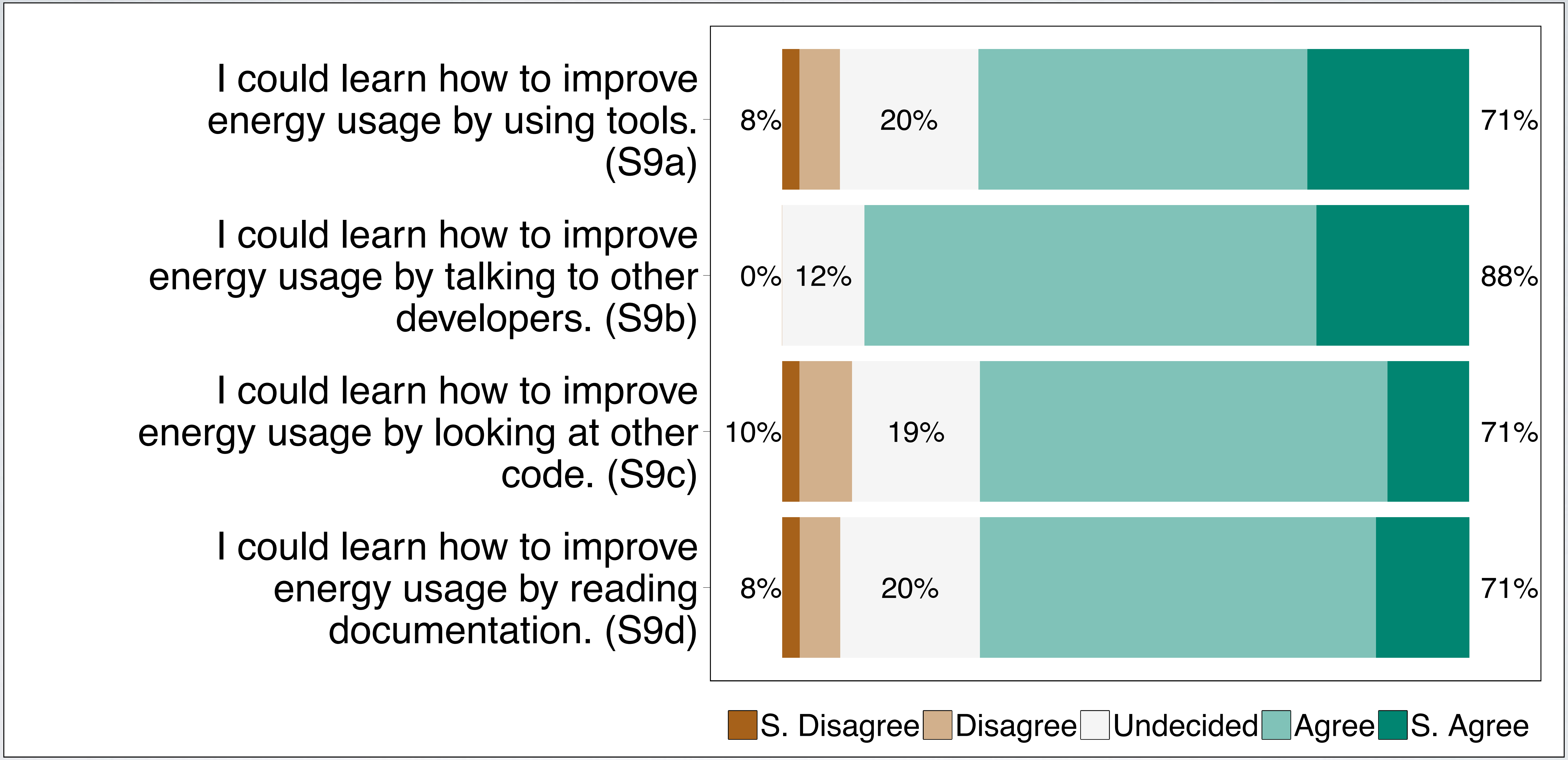
Practitioners believe that they do not have accurate intuitions about the energy usage of their code



34% believe that they have accurate intuitions but the majority either disagree (19%) or are undecided (47%).

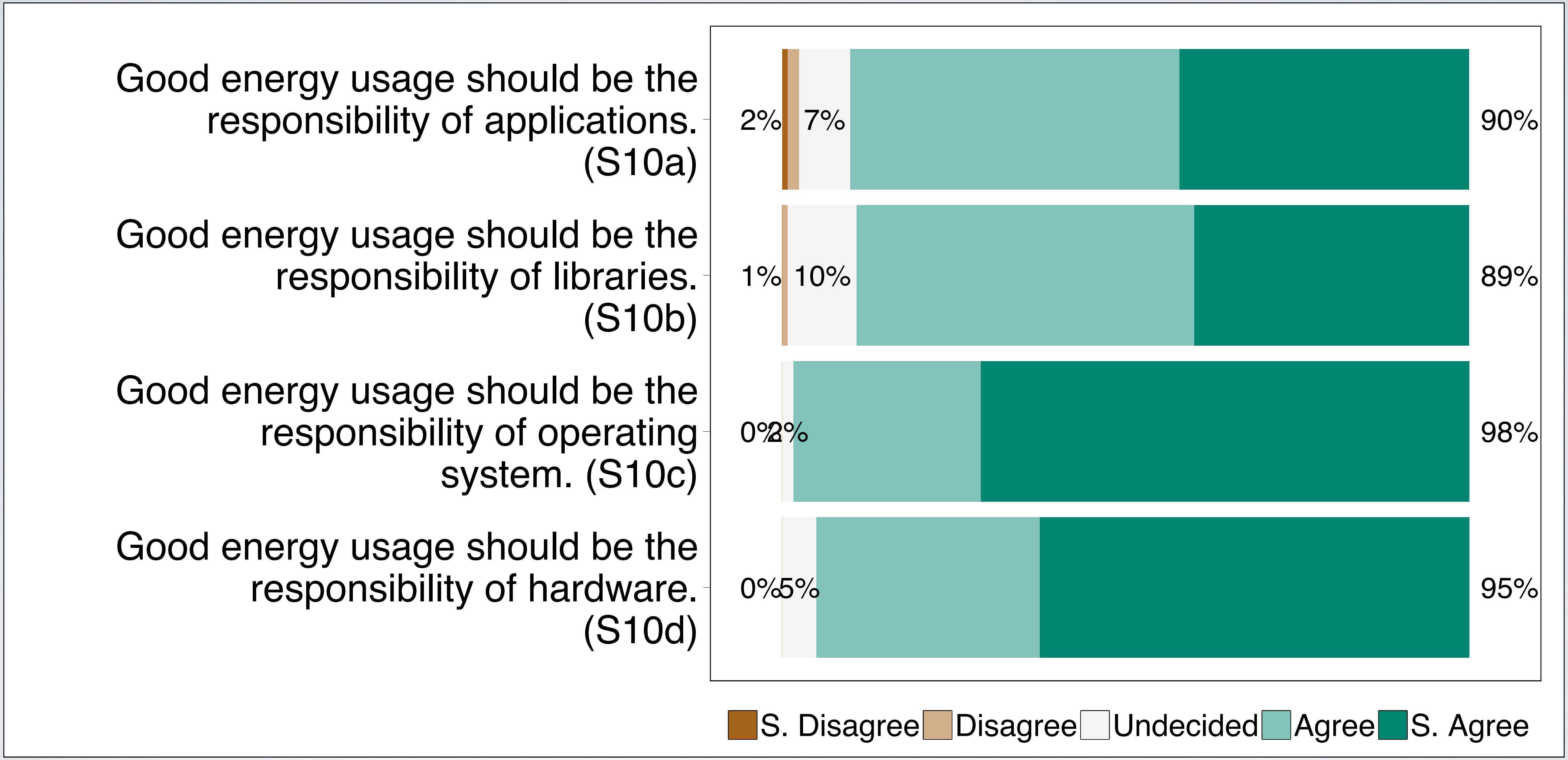
"I care about memory usage, CPU usage, I understand those. I don't have the same intuition about energy."

Practitioners believe that they could learn how to improve energy efficiency in many ways



"I would love to have more education for designing and investigating battery life time! Anything to help raise awareness and break through attitude barriers."

Energy usage should be a shared responsibility



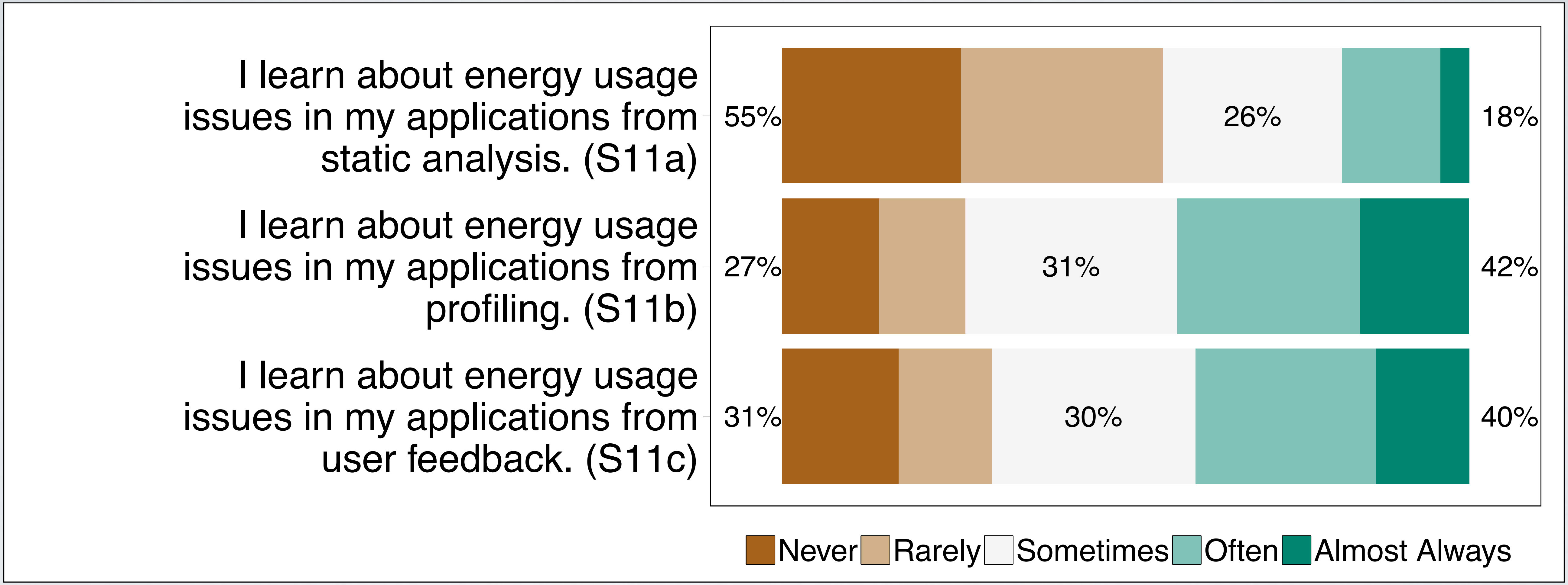
Only 3% Strongly Disagree or Disagree that applications are responsible and even less for libraries, operating systems, and hardware.

"We're all in the same boat."

PERSPECTIVES ON TESTING

- How do practitioners *currently* learn about energy issues?
- How would practitioners like to learn about energy issues?
- How difficult are energy issues to discover, diagnose, and fix?

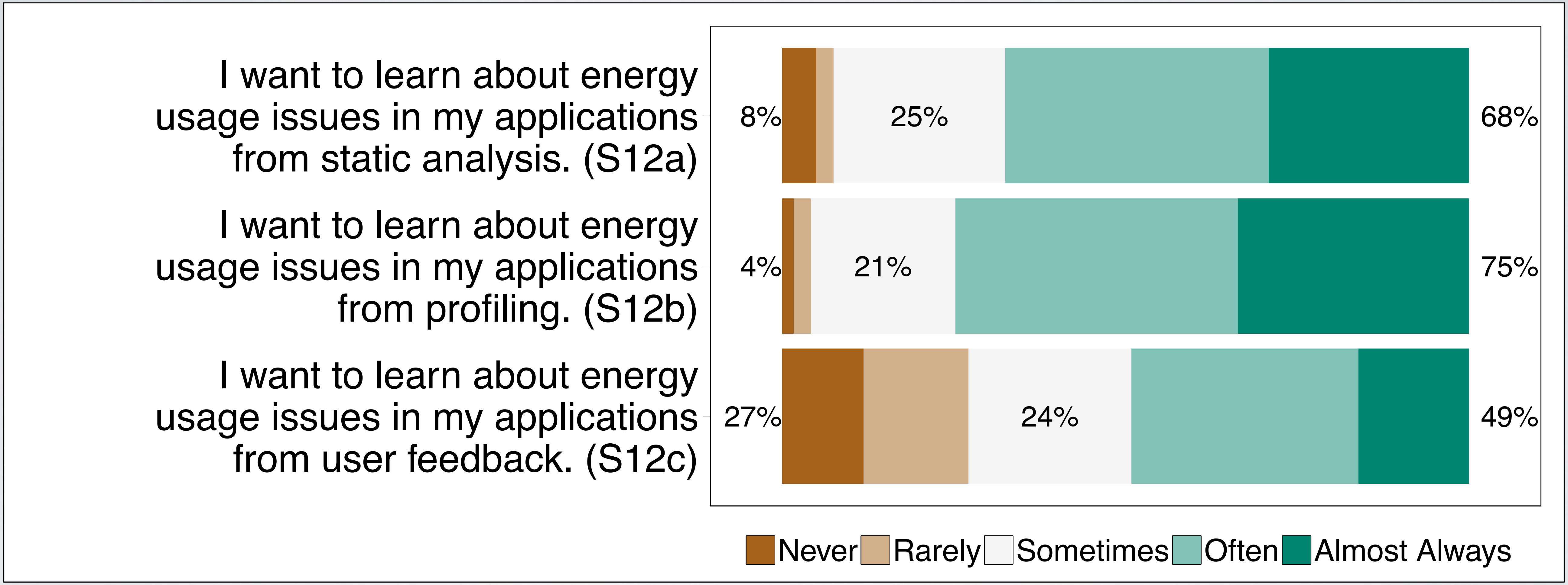
Practitioners learn about energy issues primarily from profiling and user feedback



- 73% learn from profiling Sometimes, Often, or Almost Always
- 70% learn from user feedback Sometimes, Often, or Almost Always

"Static analysis" was actually dynamic analysis or looking for code patterns, not energy issues directly.

Practitioners want to learn about energy issues primarily from profiling and static analysis

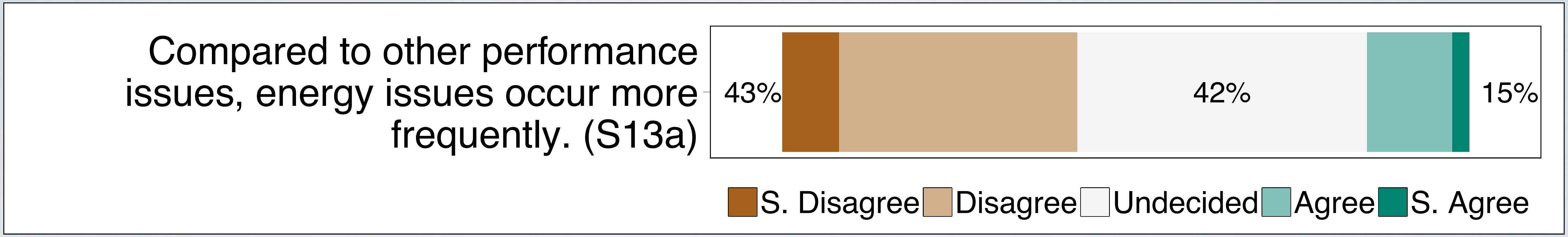


- 96% want to use profiling Sometimes, Often, or Almost Always.
- 93% want to use static analysis Sometimes, Often, or Almost Always.

"Having static analysis to point out deficiencies of efficiency would be awesome."

"Good luck getting static analysis to work on this."

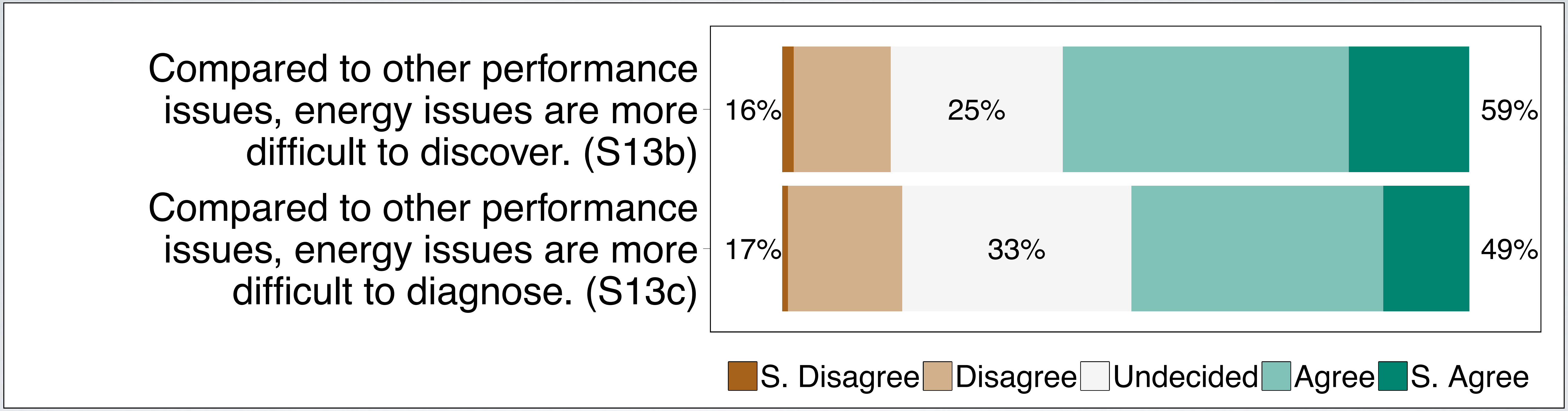
Practitioners suspect that energy issues do not occur more frequently than other performance issues



43% Strongly Disagree or Disagree but 42% are Undecided.

Missing detection capability may result in only the most egregious problems being detected.

Practitioners believe that energy issues are more difficult to discover and diagnose than performance issues



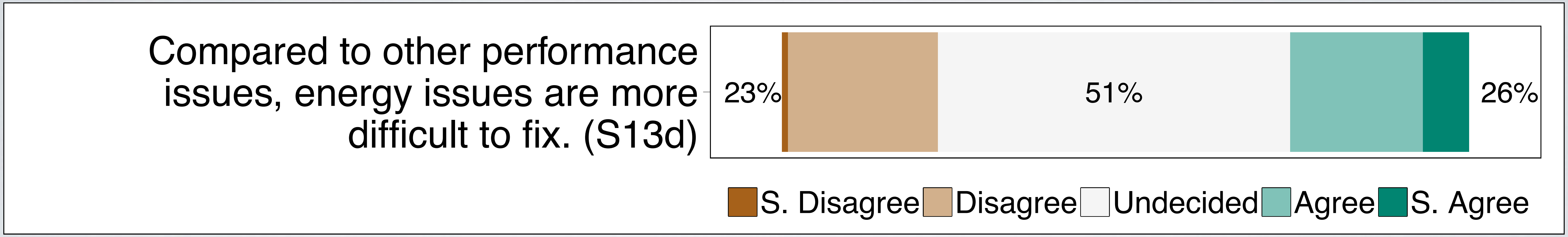
"I'm not sure what tools exist to detect such issues."

"Too many variables affect power."

"Performance issues are very obvious—the application is slow, frozen, etc.—but battery drain is a slower change and is not as immediately noticeable."

"The problem is far removed from the source and energy issues are often arise from complex interactions."

Practitioners are undecided about whether energy issues are more difficult to fix than performance issues



Most practitioners may not have fixed enough energy issues to form an overall impression of their difficulty.

Practitioners who believe that energy issues are more difficult to fix:

"If energy was not considered from the start, improving battery life could require large changes."

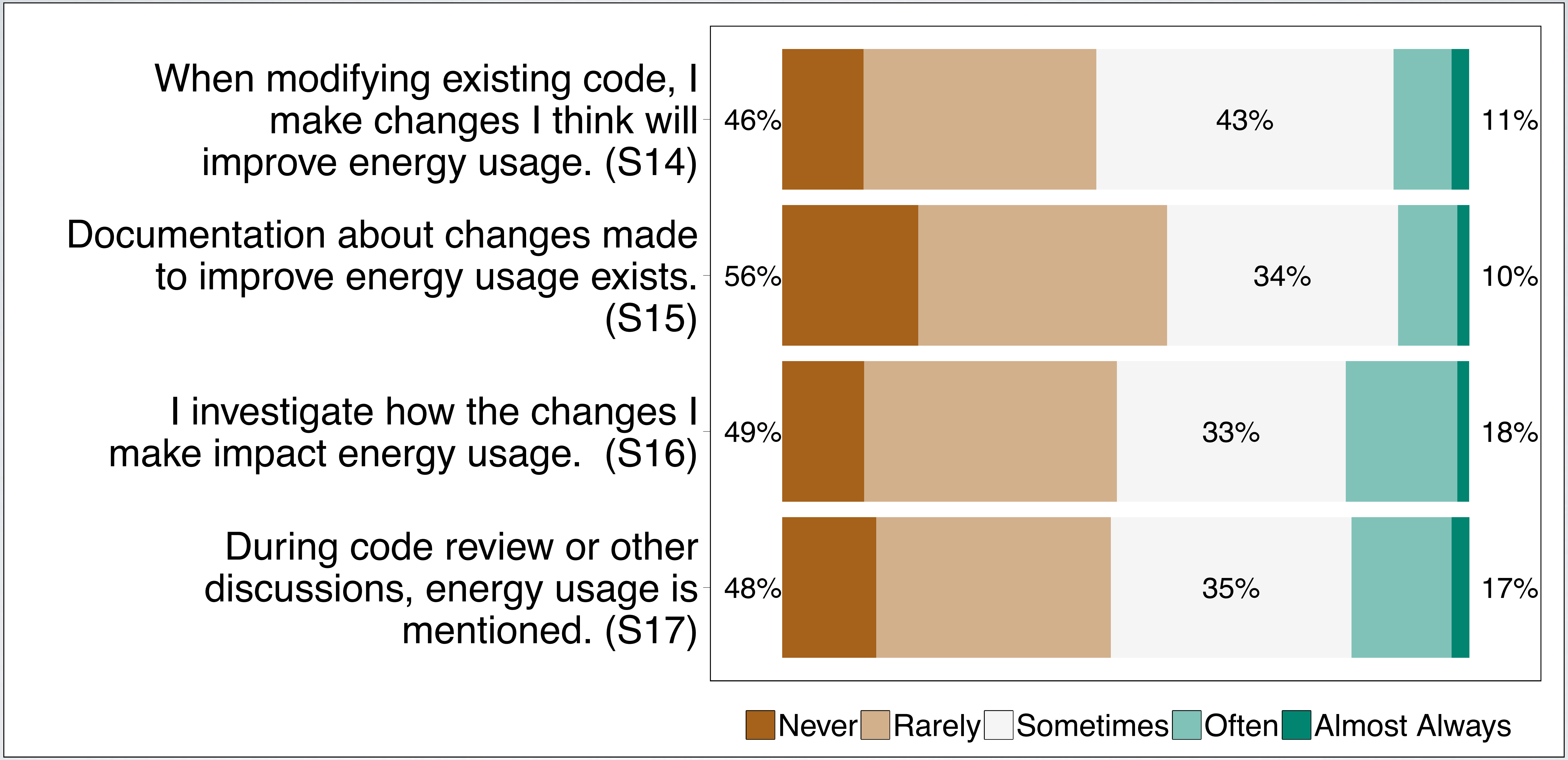
"Dependencies on libraries that are inherently inefficient can make battery life issues hard to improve."

"Problems don't always reside in the app code. The hardware often doesn't support modern commands to minimize energy usage."

PERSPECTIVES ON MAINTENANCE

- How do practitioners take energy concerns into consideration when making changes and documenting and reviewing code?

Energy concerns are largely ignored during maintenance



The largest number of respondents always chose Never or Rarely.

Why aren't practitioners documenting or discussing energy?

CONCLUSIONS

Practitioners care about energy but are not as effective as they could be at creating efficient applications

- **Requirements:** Better measurement tools could lead to more precise specifications; programming paradigms for delaying/bundling computation; techniques for managing tradeoffs between requirements.
- **Design**: Studies to validate practitioners' assumptions and provide context for decision making; tools should be scenario aware
- **Construction**: Education in any and all forms
- **Testing**: Oracles for early detection of energy issues; debugging technique that taking into account large distances between failures and root causes
- **Maintenance**: Studies to understand why developers are not discussing or documenting energy even though they believe that such actives are helpful