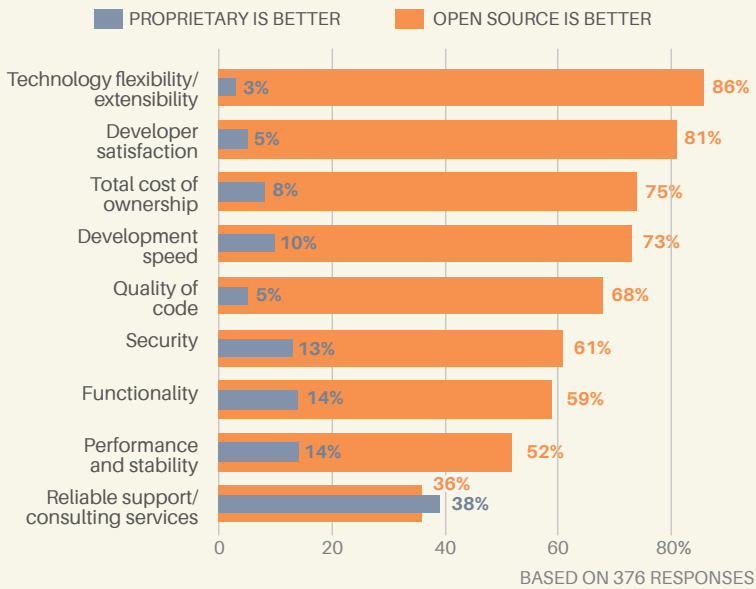# Making open source work better for developers

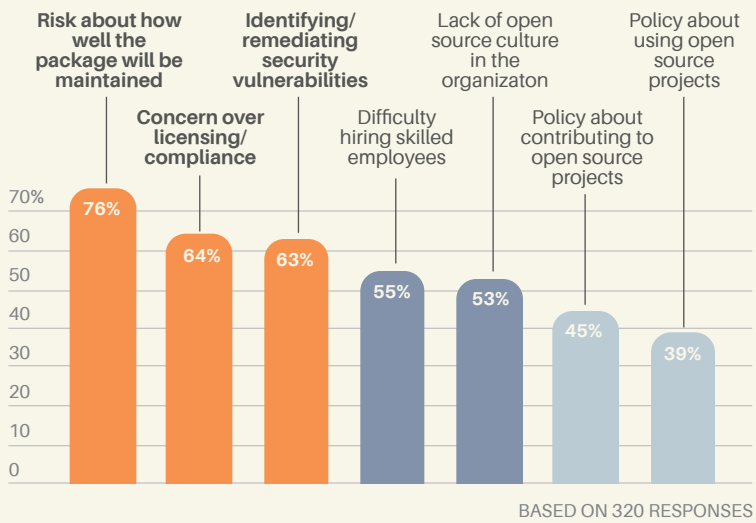HIGHLIGHTS OF THE 2019 TIDELIFT MANAGED OPEN SOURCE SURVEY

COMPARING OPEN SOURCE AND PROPRIETARY SOFTWARE

Open source is overwhelmingly seen by developers as better than proprietary software in every category except reliable support and consulting services. Closing this support gap is the last hurdle for open source to cross to completely eclipse proprietary software.

■ PROPRIETARY IS BETTER   ■ OPEN SOURCE IS BETTER

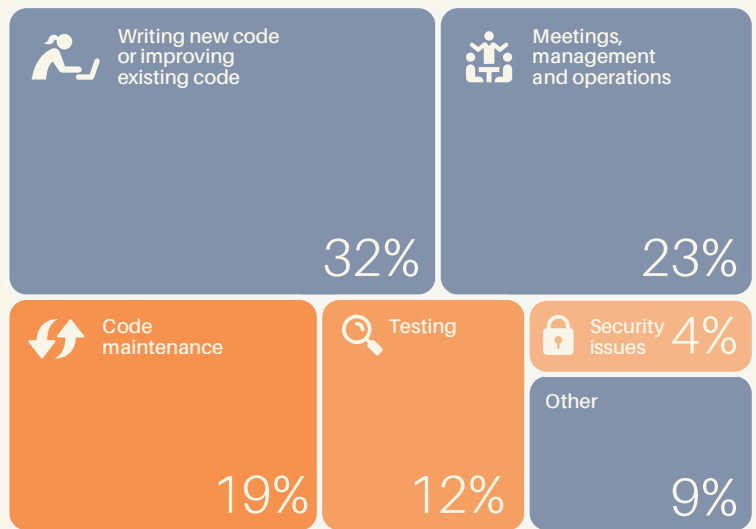| Category	Proprietary	Open Source
Technology flexibility/extensibility	3%	86%
Developer satisfaction	5%	81%
Total cost of ownership	8%	75%
Development speed	10%	73%
Quality of code	5%	68%
Security	13%	61%
Functionality	14%	59%
Performance and stability	14%	52%
Reliable support/consulting services	38%	36%

BASED ON 376 RESPONSES

OBSTACLES TO INCREASING USE OF OPEN SOURCE

Most developers are concerned about how well open source packages will be maintained into the future. Other pressing issues like concerns over licensing and compliance, identifying and remediating security vulnerabilities, and hiring skilled employees also weigh heavily on their minds.

- Risk about how well the package will be maintained — 76%
- Concern over licensing/compliance — 64%
- Identifying/remediating security vulnerabilities — 63%
- Difficulty hiring skilled employees — 55%
- Lack of open source culture in the organizaton — 53%
- Policy about contributing to open source projects — 45%
- Policy about using open source projects — 39%
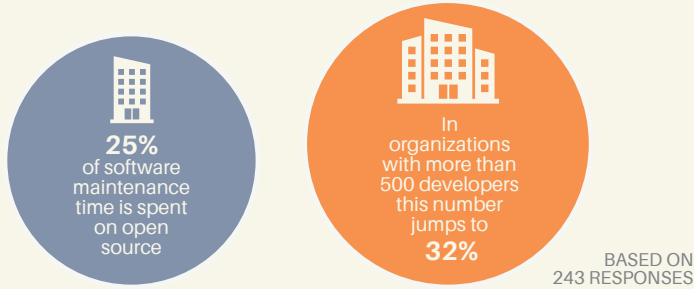
BASED ON 320 RESPONSES

HOW DEVELOPERS SPEND THEIR TIME

Developers spend more time maintaining, testing, and securing existing code than they do writing or improve code. What more could they do if they were spending less time on maintenance?
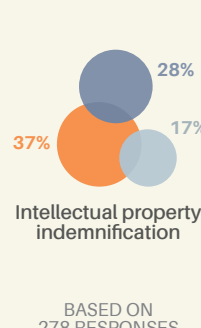
- Writing new code or improving existing code — 32%
- Meetings, management and operations — 23%
- Code maintenance — 19%
- Testing — 12%
- Security issues — 4%
- Other — 9%

BASED ON 295 RESPONSES
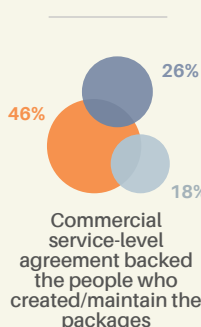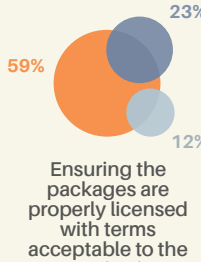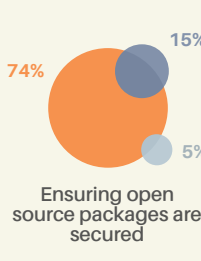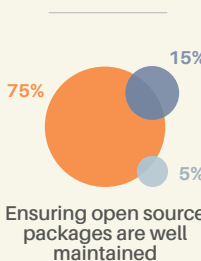
MAINTENANCE TIME RELATED TO OPEN SOURCE

Open source-related maintenance accounts for 1/4 of developers' maintenance time. In larger companies with over 500 developers, they spend almost 1/3 of their maintenance time on open source.

25% of software maintenance time is spent on open source

In organizations with more than 500 developers this number jumps to **32%**

BASED ON 243 RESPONSES

KEY BENEFITS OF MANAGED OPEN SOURCE

Developers want to know that there is someone standing behind the packages they use. A managed open source subscription gives them some of the key maintenance, security, and licensing assurances they seek.
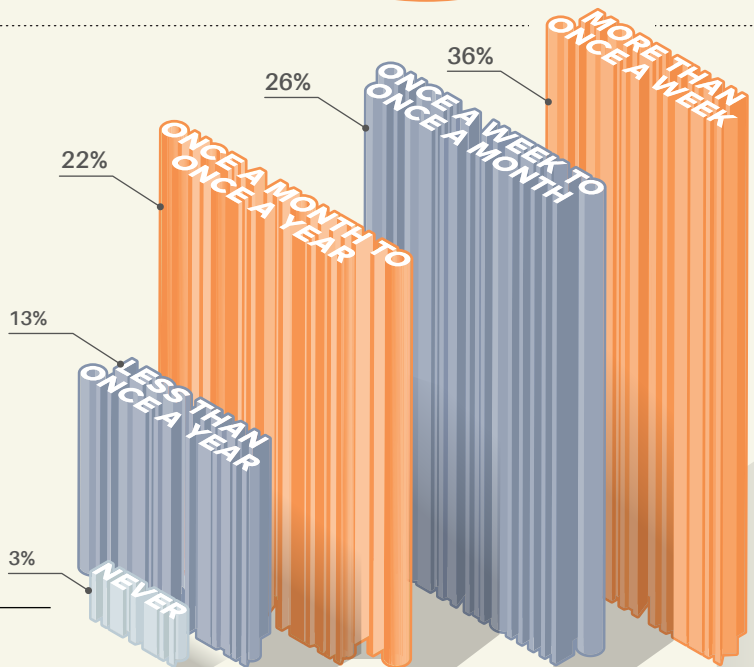
● High value
● Average value
● Low or no value

Ensuring open source packages are well maintained — 75% / 15% / 5%

Ensuring open source packages are secured — 74% / 15% / 5%

Ensuring the packages are properly licensed with terms acceptable to the organization — 59% / 23% / 12%

Commercial service-level agreement backed the people who created/maintain the packages — 46% / 26% / 18%

Intellectual property indemnification — 37% / 28% / 17%

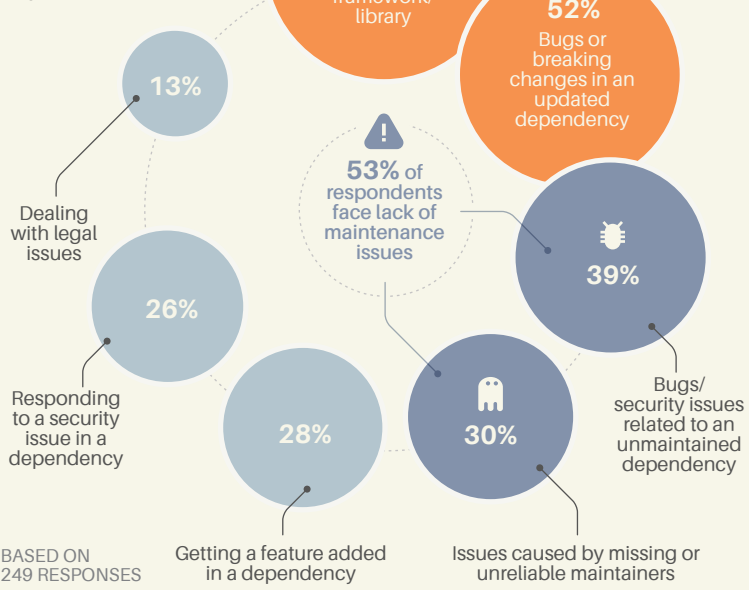BASED ON 278 RESPONSES

HOW OFTEN DO DEVELOPERS CONTRIBUTE TO OPEN SOURCE?

Most developers contribute to open source projects, with 84% contributing at least once per year.

BASED ON 286 RESPONSES

- MORE THAN ONCE A WEEK — 36%
- ONCE A WEEK TO ONCE A MONTH — 26%
- ONCE A MONTH TO ONCE A YEAR — 22%
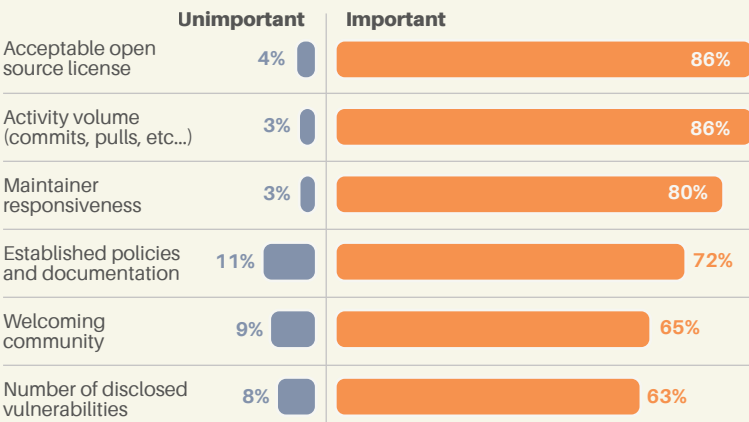- LESS THAN ONCE A YEAR — 13%
- NEVER — 3%

BIGGEST OPEN SOURCE MAINTENANCE-RELATED CHALLENGES

Time-consuming code maintenance activities create challenges, often due to issues associated with poorly maintained open source dependencies.

- Move to a new major version of a framework/library — 59%
- Bugs or breaking changes in an updated dependency — 52%
- Bugs/security issues related to an unmaintained dependency — 39%
- Issues caused by missing or unreliable maintainers — 30%
- Getting a feature added in a dependency — 28%
- Responding to a security issue in a dependency — 26%
- Dealing with legal issues — 13%

53% of respondents face lack of maintenance issues

BASED ON 249 RESPONSES

KEY FACTORS WHEN CHOOSING OPEN SOURCE PACKAGES

Project activity, licensing, and maintainer responsiveness are key factors in choosing open source projects.

Factor	Unimportant	Important
Acceptable open source license	4%	86%
Activity volume (commits, pulls, etc…)	3%	86%
Maintainer responsiveness	3%	80%
Established policies and documentation	11%	72%
Welcoming community	9%	65%
Number of disclosed vulnerabilities	8%	63%

BASED ON 320 RESPONSES

MOST IMPORTANT METRICS FOR EVALUATING OPEN SOURCE PROJECTS

Days since last activity is the most important thing developers look at when evaluating open source projects. But they also want to choose packages that are recommended by a trusted source.

- Days since last activity — 74%
- Recommended by trusted source — 61%
- Number of contributors — 54%
- Maintainer reputation — 51%
- Number of commits — 25%
- Number of clones or forks — 21%

BASED ON 303 RESPONSES