

# Open source risks: perception & mitigation

Petr Špaček, Victoria Risk

2024-05-23

[pspacek@isc.org](mailto:pspacek@isc.org), [vicky@isc.org](mailto:vicky@isc.org)



# Experimental session

- Survey
- BIND 9 DNS server
  - as an example project
- Discussion

# Survey

[https://ec.europa.eu/eusurvey/publication/  
RIPE88OpenSourceWGSurvey](https://ec.europa.eu/eusurvey/publication/RIPE88OpenSourceWGSurvey)

# Survey

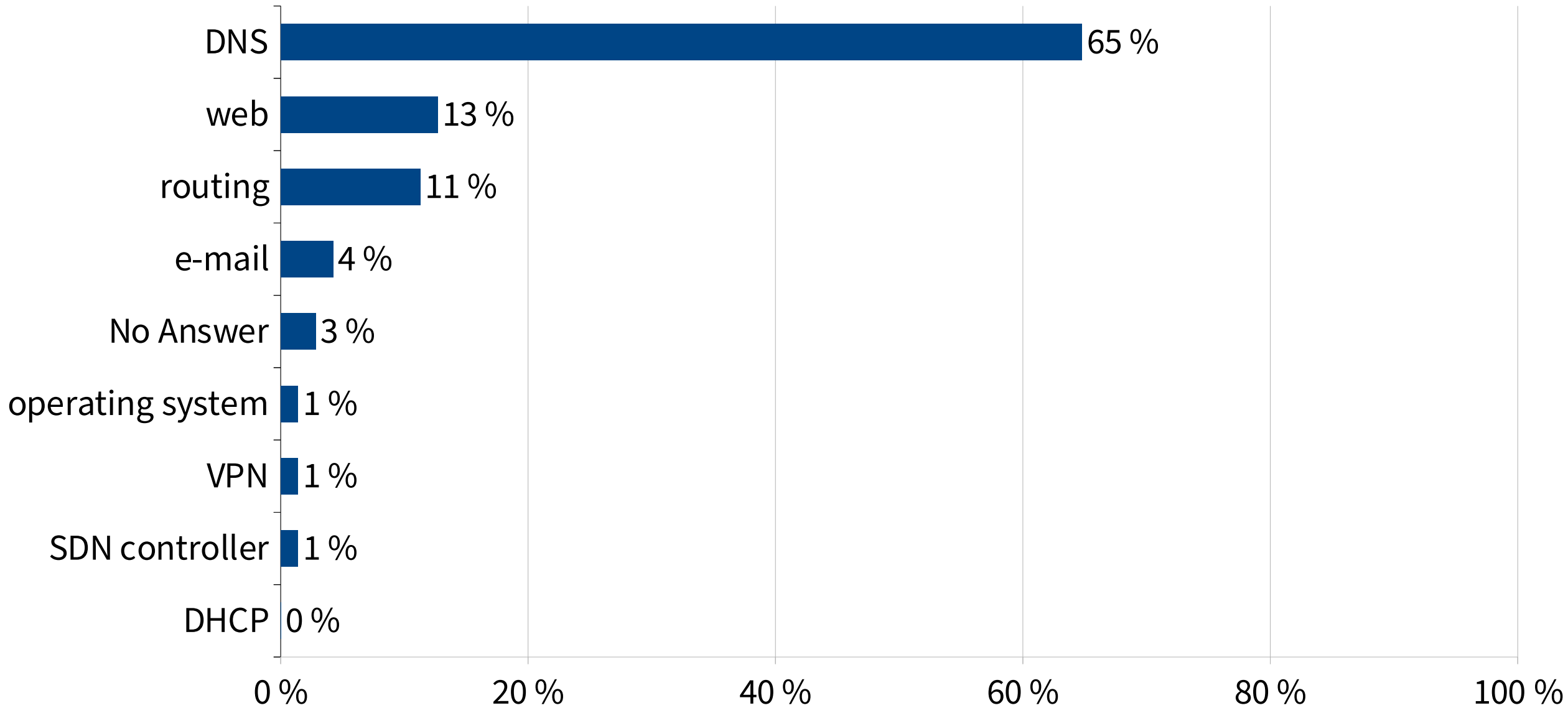
- What makes a project trustworthy?
  - "Software you consider **mission critical** in your deployment"
- Secure deployment practices
- Risk mitigation practices

# Survey

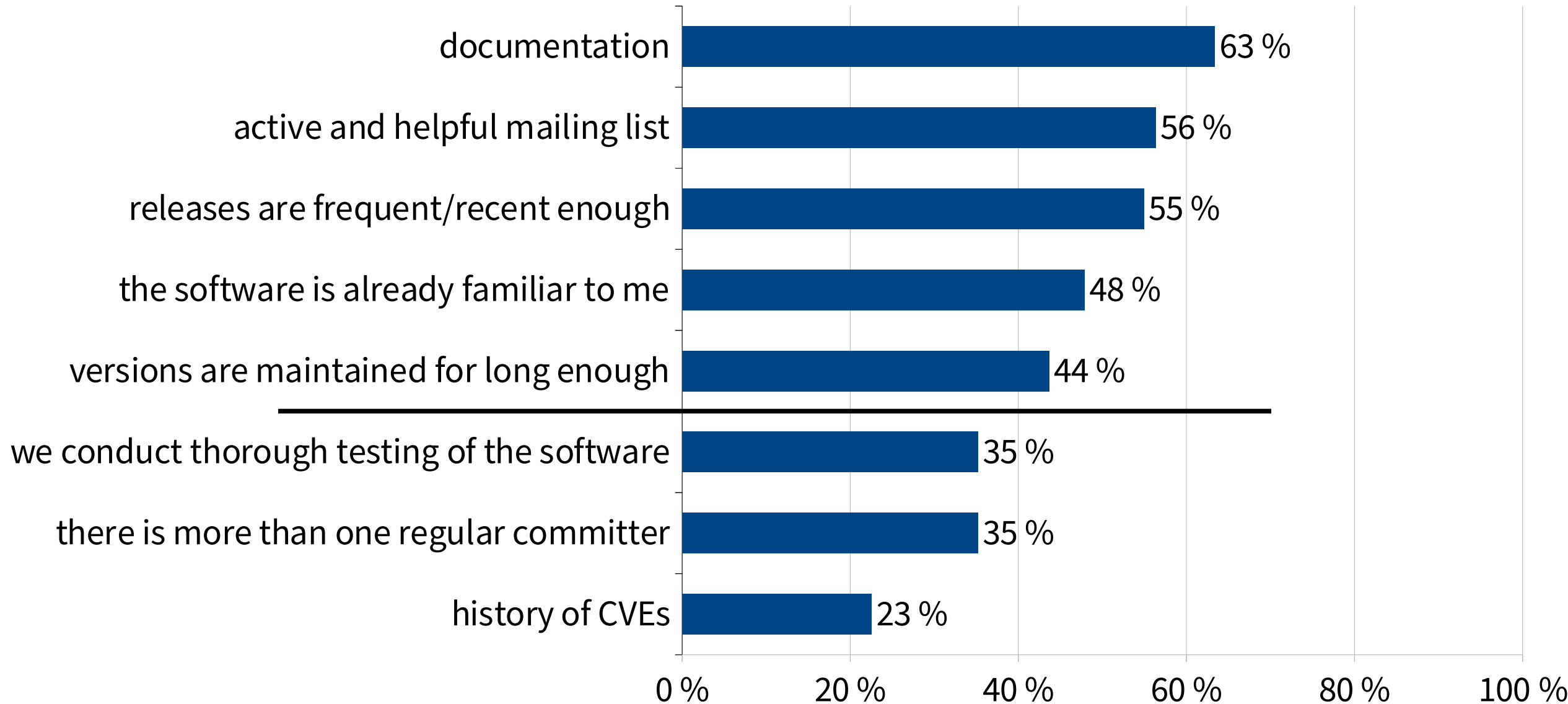
- Audience – bias
- Operators "who care"
- Presumably experts
  - RIPE Open source WG
  - RIPE DNS WG
  - dns-operations list @ DNS-OARC
  - Internet Systems Consortium's public channels
- 71 answers



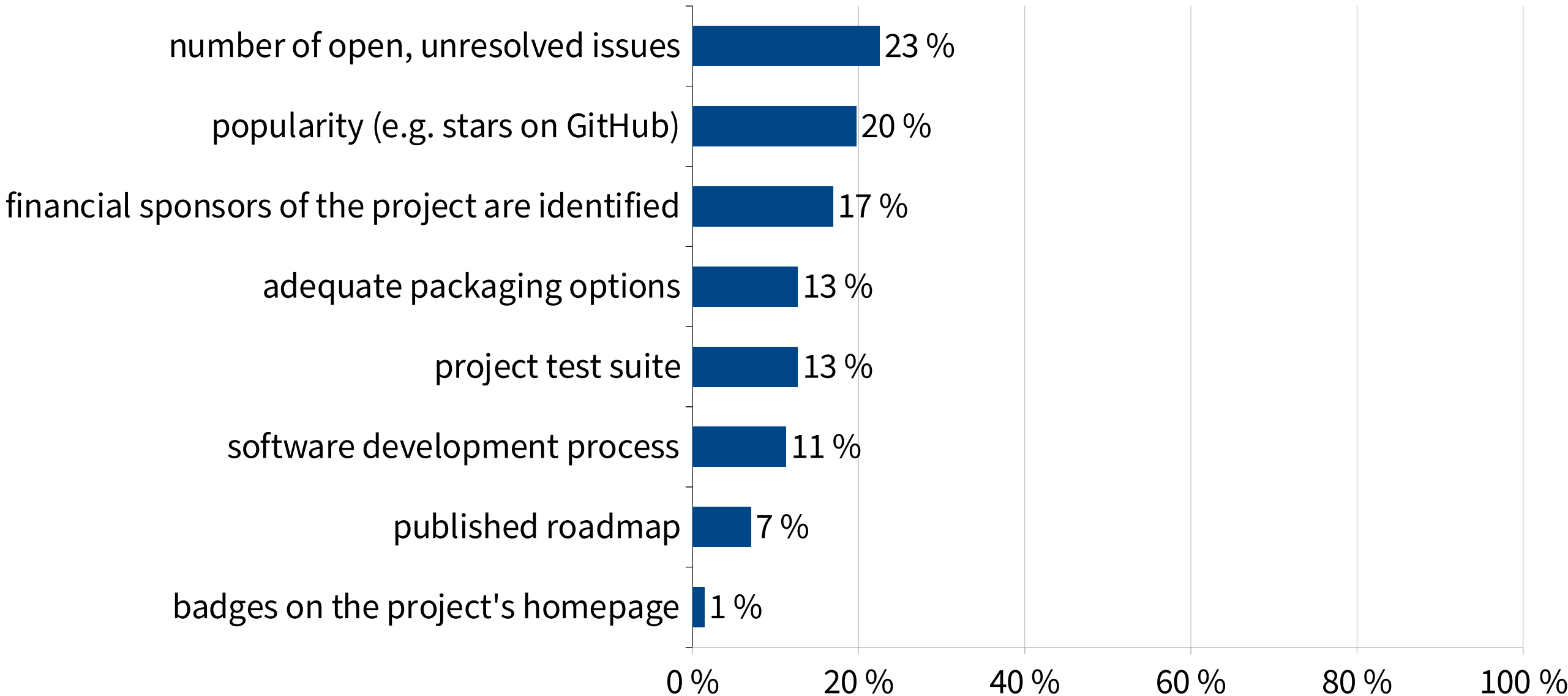
# Mission critical software



# How do you build confidence? #1

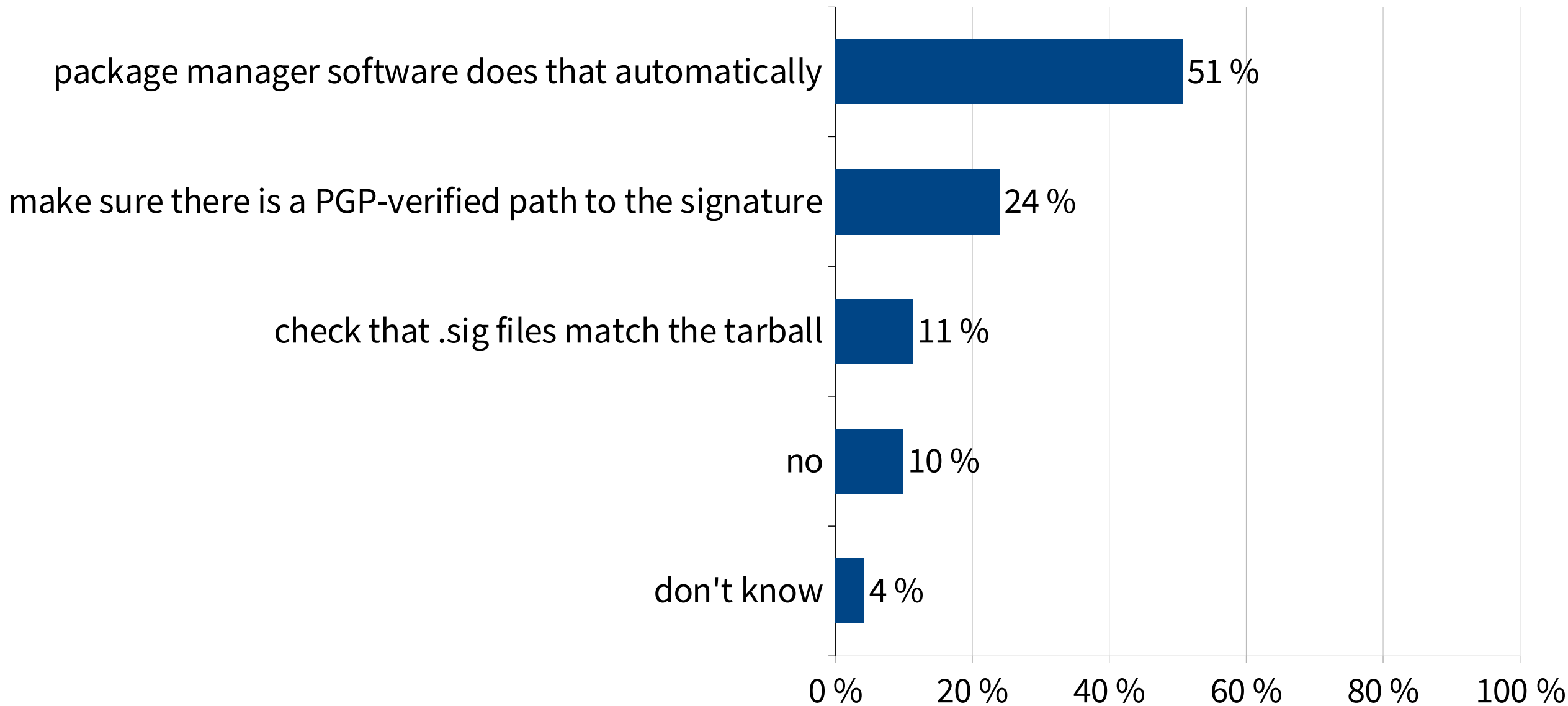


# How do you build confidence? #2

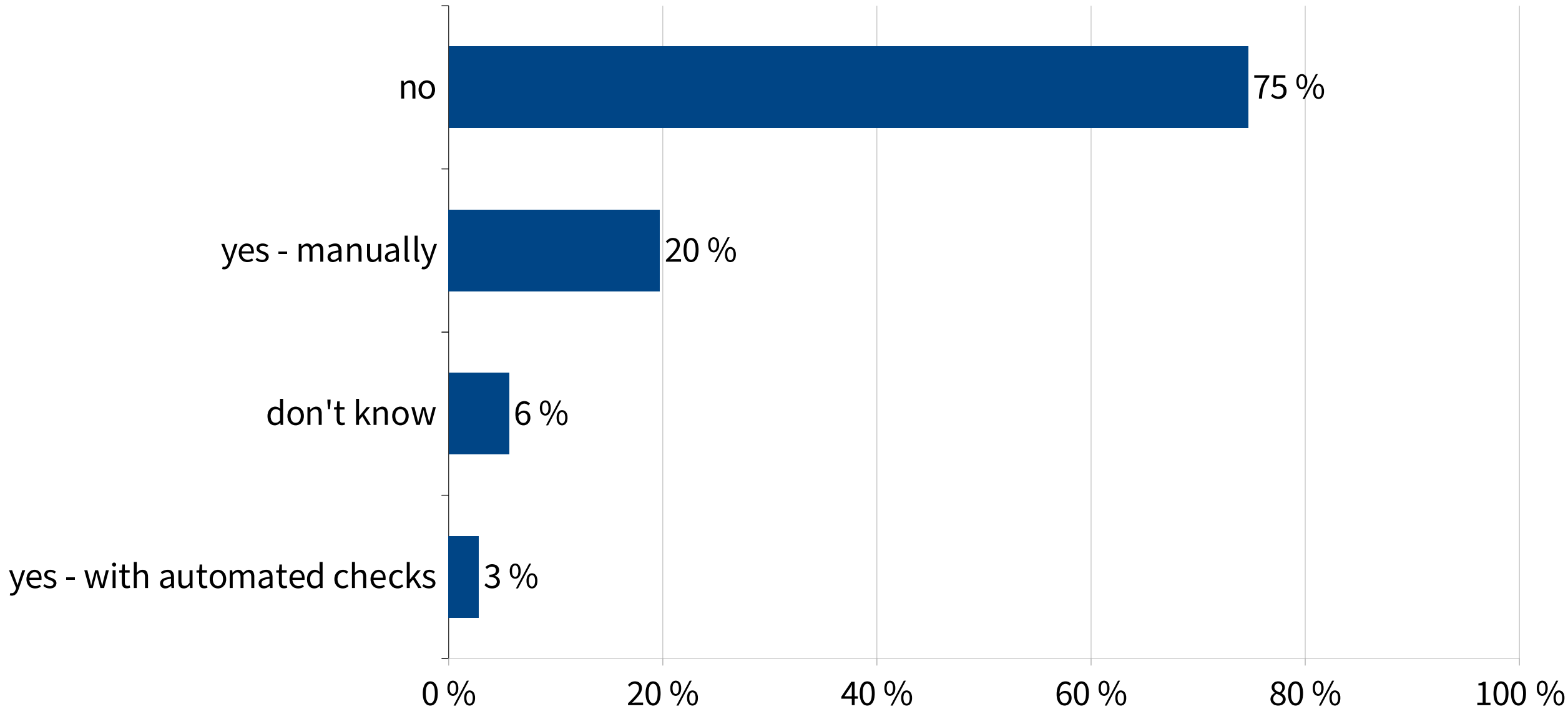




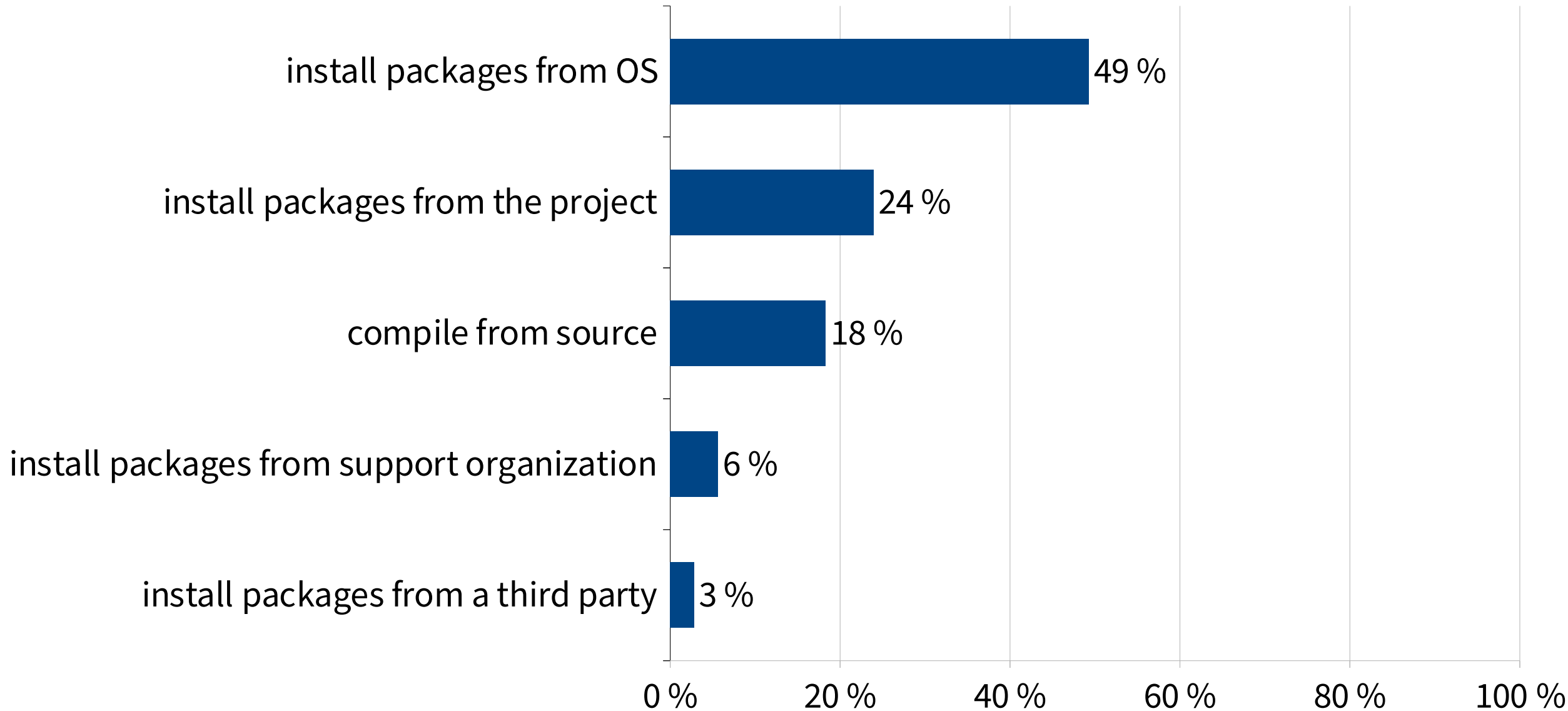
# How do you verify signatures?



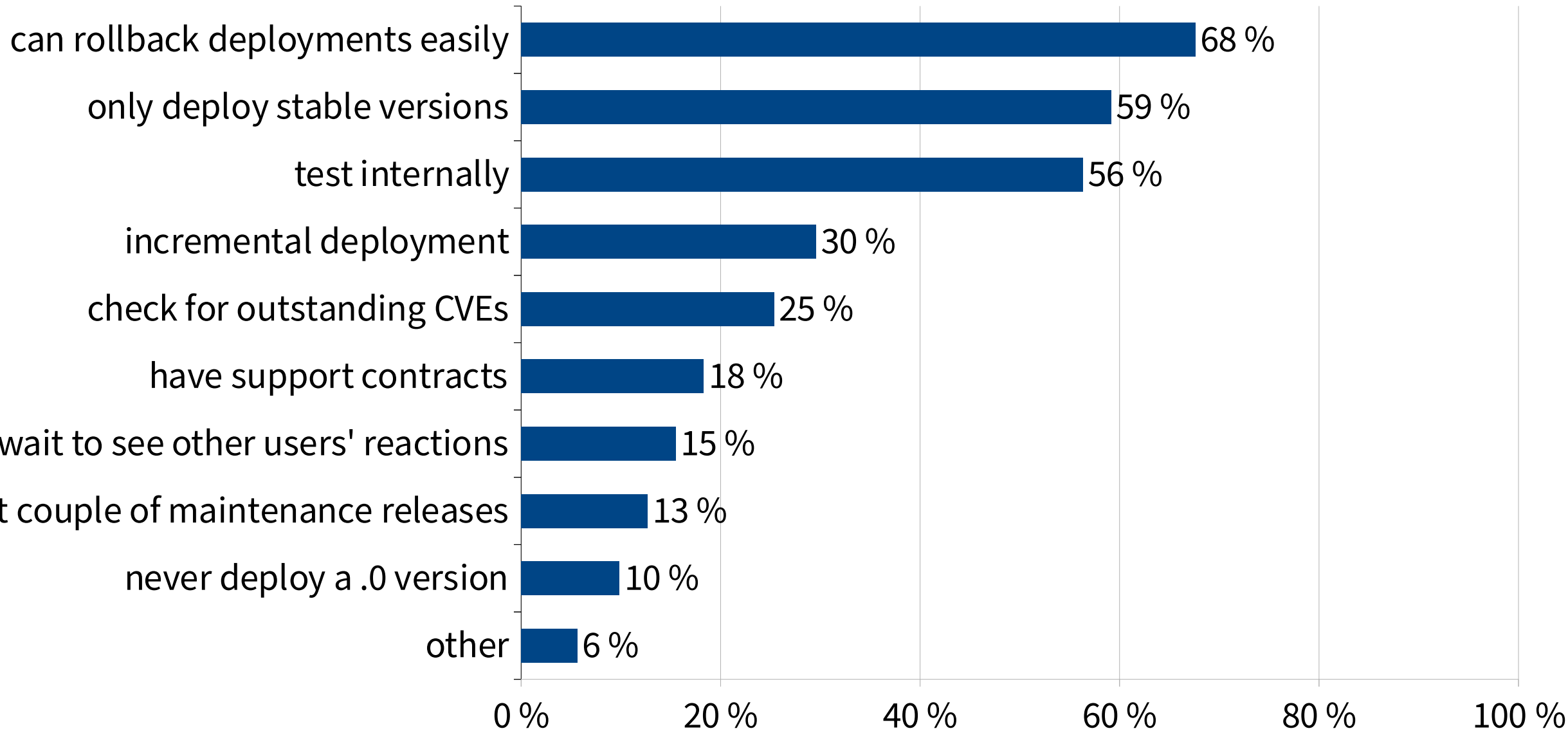
# Do you inspect source code?



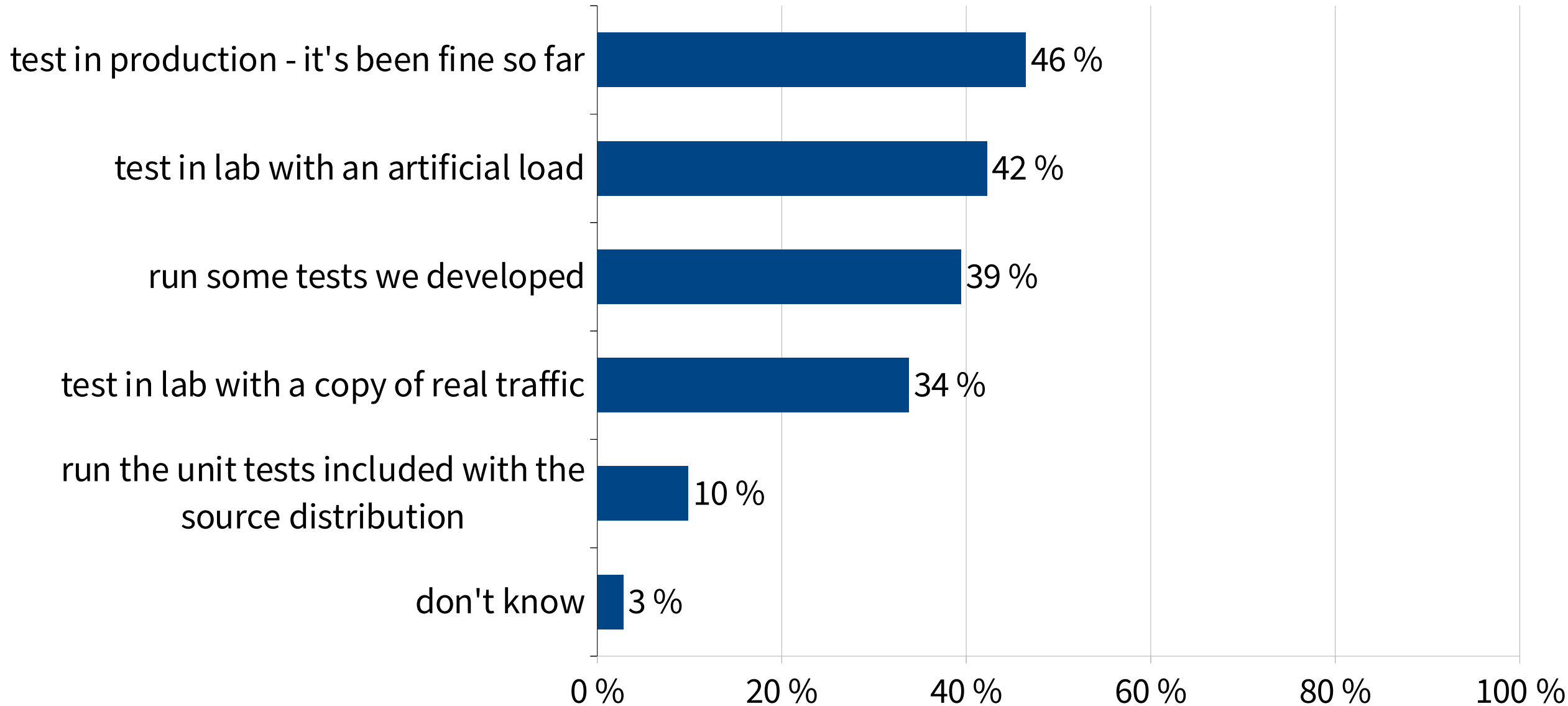
# How do you install software?



# How do you mitigate upgrade risks?



# How do you test before production?





A DNS server – example project

# BIND 9 in numbers

First commit	1998	26 years ago!
C code	263 000	lines (w/o tests or comments)
Automake	3 143	lines
Autoconf	1 839	lines
M4	1 626	lines
# of authors	50+	in the current codebase
# config knobs	325+	some are context dependent
# CVEs	130	mostly DoS



\* attribution of old code is hard – squash & merge model

# Existing code – audit

- Who knows what's in there?!
  - 26 years!
- Security audit in 2023
  - <https://www.isc.org/blogs/2024-bind-audit/>
  - 1 CVE, 2 medium severity, 6 low, 23 "nits" ...
    - Low-level bugs

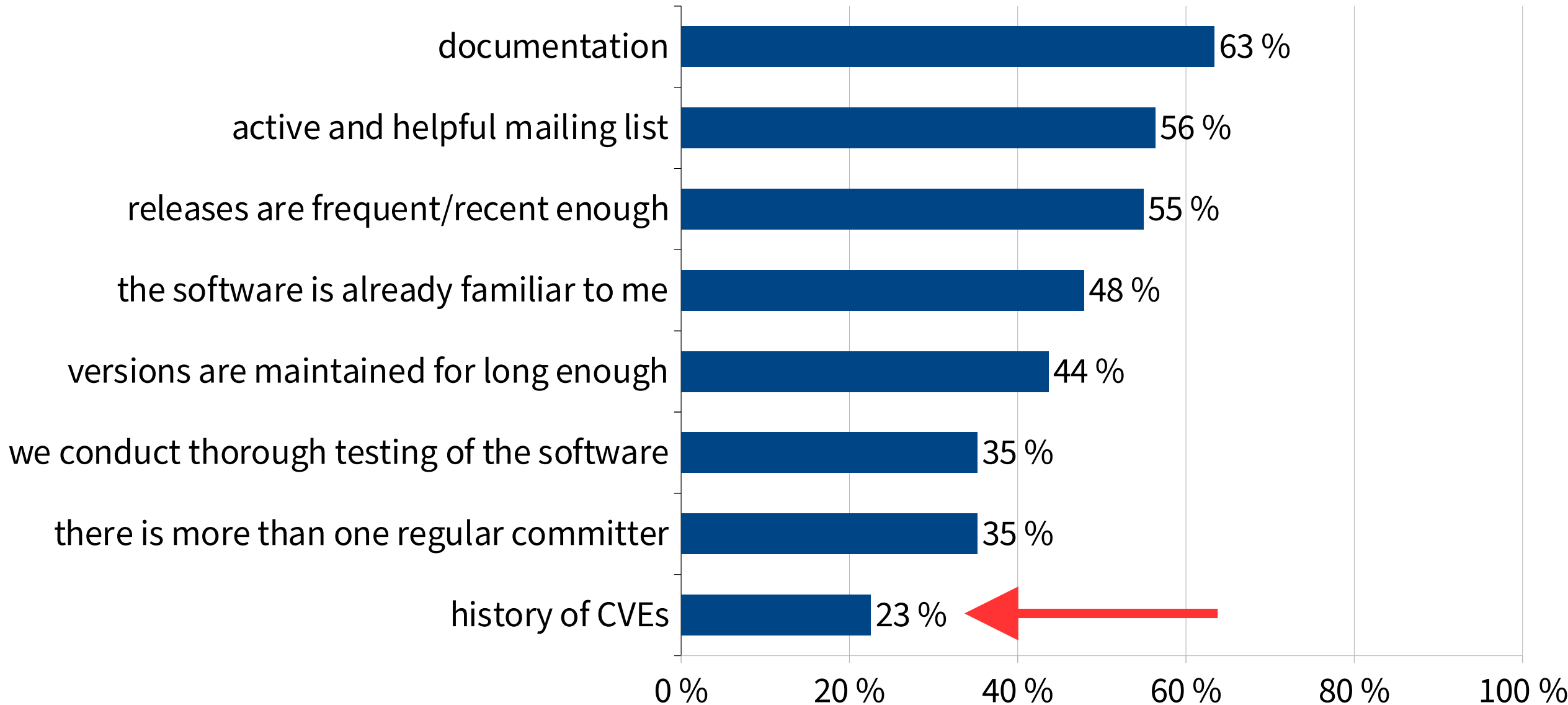


# Audit limits


- No DNS-protocol level bugs found
  - By auditors – non-DNS experts
    - Meanwhile ...

CVE #	Short Description
2023-50868	Preparing an NSEC3 closest encloser proof can exhaust CPU resources
2023-50387	KeyTrap - Extreme CPU consumption in DNSSEC validator
2023-6516	Specific recursive query patterns may lead to an out-of-memory condition
2023-5680	Cleaning an ECS-enabled cache may cause excessive CPU load
2023-5679	Enabling both DNS64 and serve-stale may cause an assertion failure ...
2023-5517	Querying RFC 1918 reverse zones may cause an assertion failure when ...
2023-4408	Parsing large DNS messages may cause excessive CPU load

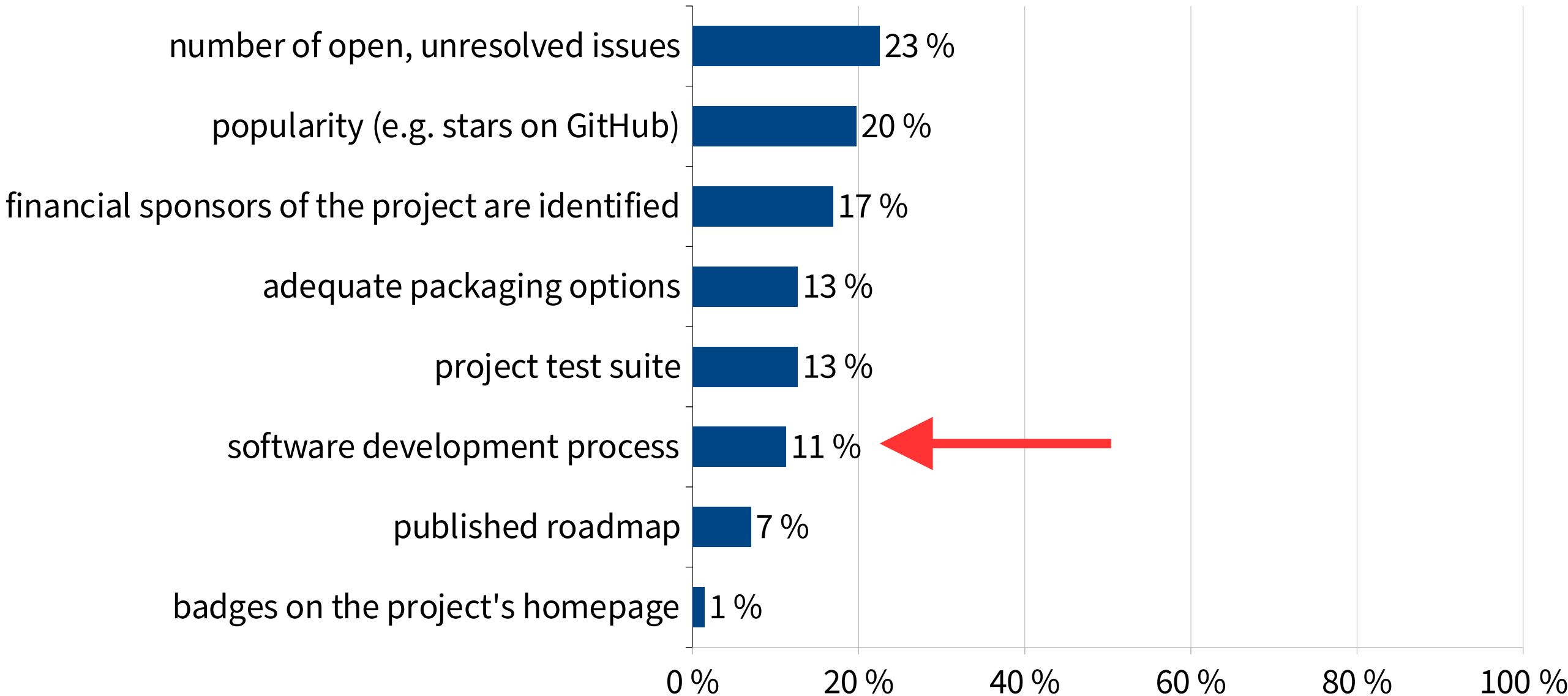
# BIND 9 vs. survey – CVEs



# Self-imposed policies

- Coding & review procedures
- OpenSSF software quality badge 
  - Lots of non-technical requirements
- ISC software defect and security vulnerability disclosure
- ISC CVSS scoring guidelines
- A lot of invisible work

# BIND 9 vs. survey – processes



# BIND 9 new code

- Peer review in GitLab
  - Very few external contributions
- Automated tests
  - Continuous integration in GitLab
  - Extra things "on side"

# Automated tests

- Unit tests
- Integration
- Fuzzers
- Interoperability
- Stress
- Performance ...



## GCC Code Coverage Report

Metric	Coverage
Lines	77.1 %
Functions	85.5 %
Branches	55.6 %

incl. 12 204 assertions,  
65.3 % without them

# Continuous integration

The screenshot displays a comprehensive CI pipeline dashboard. It is organized into several columns representing different stages of the build process:

- autotest:** Contains one job, 'autotest', which is completed.
- precheck:** Lists 15 jobs such as 'black', 'changes', 'checkbashisms', etc., all marked as completed.
- build:** Lists 25 jobs including 'clangasan', 'clangbookwormamd64', 'clangbullseyeamd64', etc., all completed.
- unit:** Lists 20 jobs like 'unit:clangasan', 'unit:clangbookwormamd64', etc., all completed.
- system:** Lists 15 jobs such as 'cross-version-config-tests', 'respdft', etc., all completed.
- performance:** Lists 10 jobs including 'shotgun-dot', 'shotgun-top', 'shotgun-udp', etc., all completed.
- docs:** Lists two jobs, 'docs' and 'docs:terball', both completed.
- postcheck:** Lists three jobs: 'rock', 'gov', and 'scan-build', all completed.
- Downstream:** Contains two jobs, 'binhd-shotgun-ci' and 'binhd-shotgun-ci', both completed.
- jobgen:** Lists one job, 'resolver-shotgun-pipeline-generator', completed.
- performance (repeated):** Lists one job, 'resolver-shotgun-child-pipeline', completed.
- Downstream (repeated):** Lists one job, 'resolver-shotgun-#182271', completed.
- build (repeated):** Lists two jobs, 'binhd-401ccc707d9f54e33467f08d0659...' and 'binhd-v0.10.23', both completed.
- test:** Lists two jobs, 'main' and 'v0.10.23', both completed.
- postproc:** Lists one job, 'postproc', completed.

For **main**

**Scheduled**

🕒 112 jobs ⌚ 69 minutes 53 seconds, queued for 14 seconds

**Pipeline**

Needs

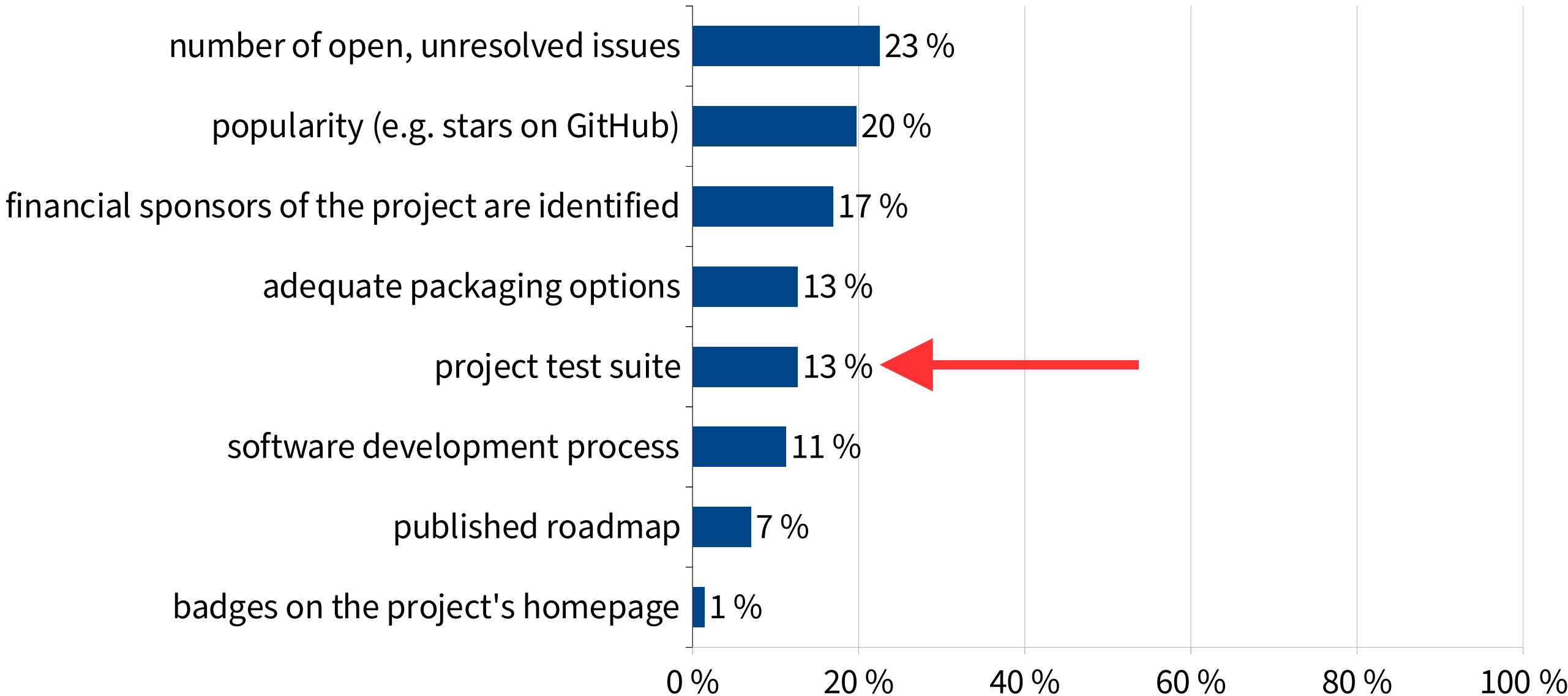
Jobs **112**

Failed Jobs **1**

Tests **6321**



# BIND 9 vs. survey – tests





# Peer review



# Peer review

16 files +141 -383

ISC Open Source Projects / BIND / Merge requests / !5071

Merged use a fixedname buffer in dns\_message\_gettempname() 2515-improve-glue-cache-pe... into main

Overview 0 Commits 2 Pipelines 6 Changes 16

Search (e.g. \*.vue) (Ctrl+P)

include/dns	
message.h	+3 -20
win32	
libdns.def.in	+0 -1
message.c	+56 -107
<b>rbtdb.c</b>	<b>+40 -129</b>
resolver.c	+0 -1
tkey.c	+4 -18
tsig.c	+3 -6
xfrin.c	+0 -2
zone.c	+0 -3
ns	
client.c	+12 -38
query.c	+8 -21
xfrouit.c	+0 -2

lib/dns/rbtdb.c

```
9927 - 0xfc, 0xfd, 0xfe, 0xff
9888 + static const unsigned char maptoupper[256] = {
9889 +   ['a'] = 'A', ['b'] = 'B', ['c'] = 'C', ['d'] = 'D', ['e'] = 'E',
9890 +   ['f'] = 'F', ['g'] = 'G', ['h'] = 'H', ['i'] = 'I', ['j'] = 'J',
9891 +   ['k'] = 'K', ['l'] = 'L', ['m'] = 'M', ['n'] = 'N', ['o'] = 'O',
9892 +   ['p'] = 'P', ['q'] = 'Q', ['r'] = 'R', ['s'] = 'S', ['t'] = 'T',
9893 +   ['u'] = 'U', ['v'] = 'V', ['x'] = 'X', ['y'] = 'Y', ['z'] = 'Z',
9928 9894 };
9937 - unsigned char bits;
9938 - unsigned char c, flip;
9901 + rdatasetheader_t *header = NULL;
9902 + uint8_t mask = (1 << 7);
9903 + uint8_t bits = 0;
9939 9904
9940 9905     header = (struct rdatasetheader *) (raw - sizeof(*header));
9941 9906
@@ -9946,85 +9911,36 @@ rdataset_getownercase(const dns_rdataset_t *rdataset, dns_name_t *name) {
```

 Approve



# Release ... err, try again ...



ISC (<https://fosstodon.org/@iscdotorg>)

@ISCdotORG




Ok, this is embarrassing. Please don't install the [@bind9](#) updated versions we posted yesterday. Someone reported an error - we left out a LETTER of the ALPHABET in a streamlined routine. We will be removing the new versions and reposting after we have a chance to retest.

5:24 AM · Jun 18, 2021

# Automated tests – limits

 ISC Open Source Projects /  BIND / Issues / #2779

## W or w characters in domain names are altered to "\000" ⋮

 Closed  Issue created 2 years ago by Sean Zhang

### Summary

We recently upgraded our bind9 from `1:9.16.16-2+ubuntu18.04.1+isc+1` to `1:9.16.17-1+ubuntu21.04.1+isc+1` and start experiencing some wildcard names not being resolved. The resolver will return `servfail`. After some troubleshooting we found that:

Under certain conditions (reproducible), the name in answer will not match the name in question. Found this issue reproducible with following conditions:



# Peer review – limits

ISC Open Source Projects / BIND / Merge requests / !5071

Merged use a fixedname buffer in dns\_message\_gettempname() 2515-improve-glue-cache-pe...

Overview 0 Commits 2 Pipelines 6 Changes 16

lib/dns/rbtdb.c

```
9900 -          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
9901 -          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
9902 -          0x00, 0x00, 0x00, 0x00
9880 + static const unsigned char maptolower[256] = {
9881 +         ['A'] = 'a', ['B'] = 'b', ['C'] = 'c', ['D'] = 'd', ['E'] = 'e',
9882 +         ['F'] = 'f', ['G'] = 'g', ['H'] = 'h', ['I'] = 'i', ['J'] = 'j',
9883 +         ['K'] = 'k', ['L'] = 'l', ['M'] = 'm', ['N'] = 'n', ['O'] = 'o',
9884 +         ['P'] = 'p', ['Q'] = 'q', ['R'] = 'r', ['S'] = 's', ['T'] = 't',
9885 +         ['U'] = 'u', ['V'] = 'v', ['X'] = 'x', ['Y'] = 'y', ['Z'] = 'z',
9903 9886     };
9904 9887
```

The matrix is square! It must be fine!

# BIND 9 Release process

## Release Checklist

### Before the Code Freeze

- ✓ (QA) Rebase -S editions on top of current open-source versions: `git checkout bind-9.18-sub && git rebase origin/bind-9.18`
- ✓ (QA) Inform Support and Marketing of impending release (and give estimated release dates).
- ✓ (QA) Ensure there are no permanent test failures on any platform. Check [public](#) and [private](#) scheduled pipelines.
- ✓ (QA) Check charts from `shotgun:*` jobs in the scheduled pipelines to verify there is no unexplained performance drop for any protocol.
- ✓ (QA) Check [Perflab](#) to ensure there has been no unexplained drop in performance for the versions being released.
- ✓ (QA) Check whether all issues assigned to the release milestone are resolved<sup>1</sup>.
- ✓ (QA) Ensure that there are no outstanding [merge requests in the private repository](#)<sup>1</sup> (Subscription Edition only).
- ✓ (QA) Ensure all merge requests marked for backporting have been indeed backported.
- ✓ (QA) Announce (on Mattermost) that the code freeze is in effect.

### Before the Tagging Deadline

- ✓ (QA) Inspect the current output of the `cross-version-config-tests` job to verify that no unexpected backward-incompatible change was introduced in the current release cycle.
- ✓ (QA) Ensure release notes are correct, ask Support and Marketing to check them as well. [Example](#)
- ✓ (QA) Add a release marker to `CHANGES`. Examples: [9.18](#), [9.10](#)
- ✓ (QA) Add a release marker to `CHANGES.SE` (Subscription Edition only). [Example](#)
- ✓ (QA) Update BIND 9 version in `configure.ac` ([9.18+](#)) or `version` ([9.10](#)).
  - (QA) Rebuild `configure` using `Autoconf` on `docs.isc.org` ([9.10](#)).
  - (QA) Update GitHub settings for all maintained branches to disallow merging to them: [public](#), [private](#)
- ✓ (QA) Tag the releases in the private repository (`git tag -s -m "BIND 9.x.y" v9.x.y`).

### Before the ASN Deadline (for ASN Releases) or the Public Release Date (for Regular Releases)

- ✓ (QA) Check that the formatting is correct for the HTML version of release notes.
- ✓ (QA) Check that the formatting of the generated man pages is correct.
- ✓ (QA) Verify GitHub CI results [for the tags](#) created and sign off on the releases to be published.
  - (QA) Update GitHub settings for all maintained branches to allow merging to them again: [public](#), [private](#)
- ✓ (QA) Prepare (using `version_bump.py`) and merge MRs resetting the release notes and updating the version string for each maintained branch.
- ✓ (QA) Rebase the Subscription Edition branches (including recent release prep commits) on top of the open source branches with updated version strings.
- ✓ (QA) Announce (on Mattermost) that the code freeze is over.
- ✓ (QA) Request signatures for the tarballs, providing their location and checksums. Ask [signers on Mattermost](#).
- ✓ (Signers) Ensure that the contents of tarballs and tags are identical.
- ✓ (Signers) Validate tarball checksums, sign tarballs, and upload signatures.
- ✓ (QA) Verify tarball signatures and check tarball checksums again: Run `publish_bind.sh` on `repo.isc.org` to pre-publish.
  - (QA) Prepare the `patches/` subdirectory for each security release (if applicable).
- ✓ (QA) Pre-publish ASN and/or Subscription Edition tarballs so that packages can be built.
- ✓ (QA) Build and test ASN and/or Subscription Edition packages (in [cloudsmith branch in private repo](#)). [Example](#)
- (Marketing) Prepare and send out ASN emails (as outlined in the CVE checklist; if applicable).

### On the Day of Public Release

- (QA) Wait for clearance from Security Officer to proceed with the public release (if applicable).
- ✓ (QA) Place tarballs in public location on FTP site.
- ✓ (QA) Inform Marketing of the release, providing FTP links for the published tarballs.
- ✓ (QA) Use the [Printing Press project](#) to prepare a release announcement email.
- ✓ (Marketing) Publish links to downloads on ISC website. [Example](#)
- ✓ (Marketing) Update the BIND -S Information document in SF with download links to the new versions. (If this is a security release, this will have already been done as part of the ASN process.)
- ✓ (Marketing) Update the Current Software Versions document in the SF portal if any stable versions were released.
- ✓ (Marketing) Send the release announcement email to the `bind-announce` mailing list (and to `bind-users` if a major release - [example](#)).
- ✓ (Marketing) Announce release on social media sites.
- ✓ (Marketing) Update [Wikipedia entry for BIND](#).
- ✓ (Support) Add the new releases to the [vulnerability matrix in the Knowledge Base](#).
- ✓ (Support) Update tickets in case of waiting support customers.
- ✓ (QA) Build and test any outstanding private packages in [private repo](#). [Example](#)
- ✓ (QA) Build public RPMs. [Example](#) `coast` which triggers `Copr` builds automatically
- ✓ (SwEng) Build Debian/Ubuntu packages.
- ✓ (SwEng) Update Docker files [here](#) and make sure push is synchronized to [GitHub](#). [Docker Hub](#) should pick it up automatically. [Example](#)
- ✓ (QA) Ensure all new tags are annotated and signed. `git show --show-signature v9.19.12`
- ✓ (QA) Push tags for the published releases to the public repository.
- ✓ (QA) Using `merge_tag.py`, merge published release tags back into their relevant development/maintenance branches.
- ✓ (QA) Ensure `allow_failure: true` is removed from the `cross-version-config-tests` job if it was set during the current release cycle.
- ✓ (QA) Sanitize confidential issues which are assigned to the current release milestone and do not describe a security vulnerability, then make them public.
- (QA) Sanitize [confidential issues](#) which are assigned to older release milestones and describe security vulnerabilities, then make them public if appropriate<sup>2</sup>.
- ✓ (QA) Update QA tools used in GitHub CI (e.g. Black, PyLint, Sphinx) by modifying the relevant [Dockerfile](#).
- ✓ (QA) Run a pipeline to rebuild all [images](#) used in GitHub CI.
- ✓ (QA) Update `metadata.json` with the upcoming release information.

# BIND 9 release process

- Check list of changes that went in (again)
- Polish docs
- Run tests (again)
- Generate tarball
  - Check reproducibility
- Sign
- Publish
- Build packages

# BIND 9 tarball checks

- Git ⇒ tarball reproducibility
  - <https://gitlab.isc.org/isc-projects/BIND-9/-/blob/main/util/release-tarball-comparison.sh>
  - 100 lines
  - easy enough for independent review

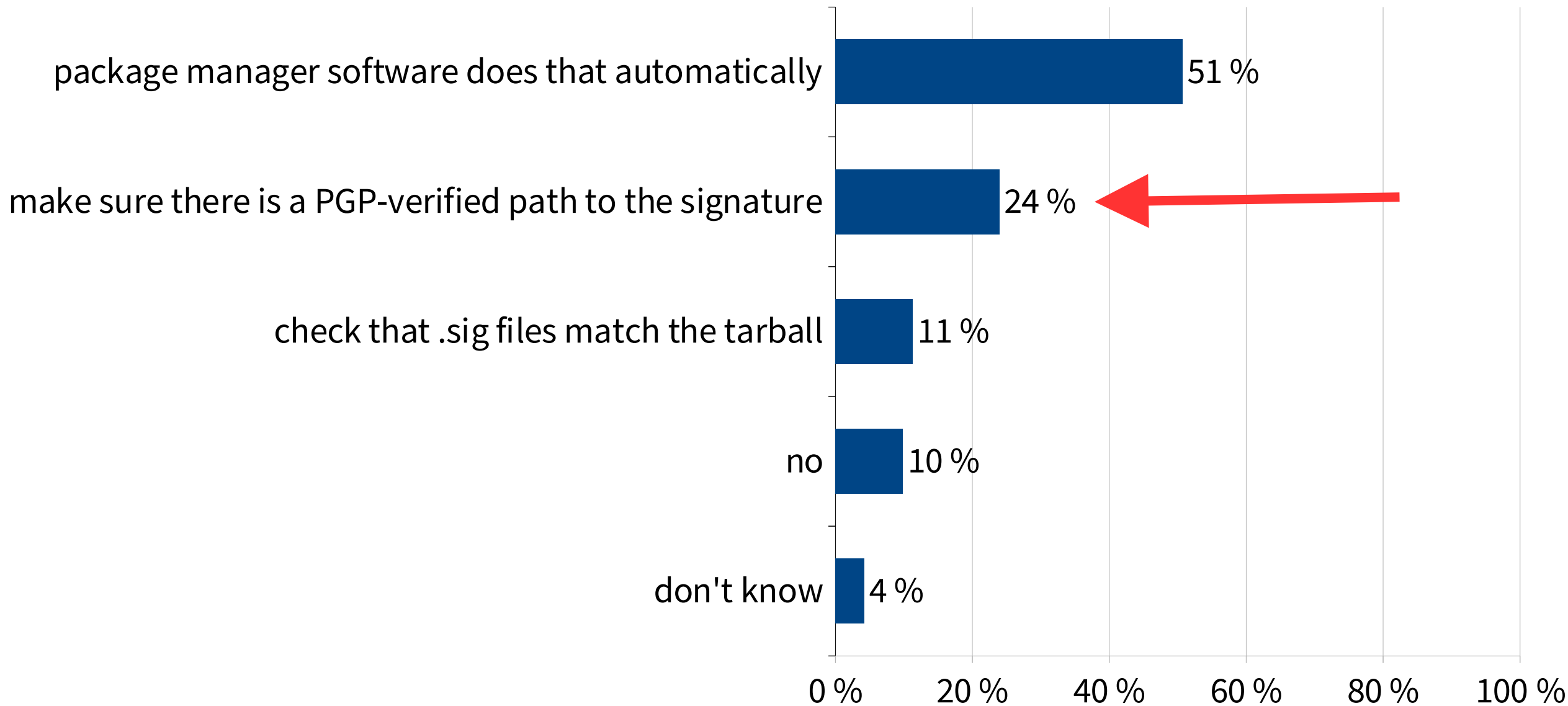


# BIND 9 tarball signing

- Dedicated VM
  - takes tarball from Gitlab
  - requests GPG signature
- Signer – person
  - SSH into the VM
  - forwards GPG agent socket



# BIND 9 vs. survey – signatures

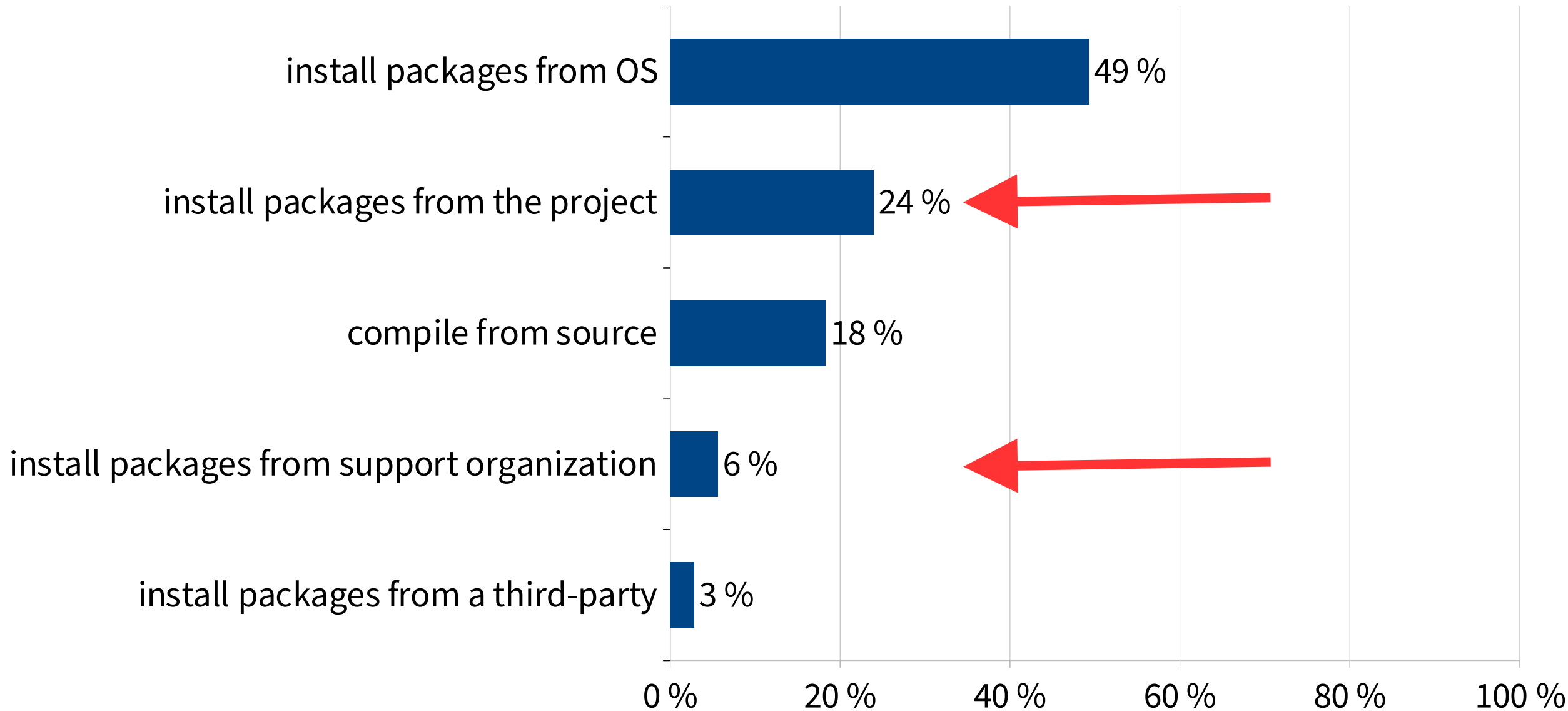


# BIND 9 package build

- Our RPM packages build in Gitlab
- Copr, Launchpad, Docker, etc. – manual

build	push	install:staging	update:staging	upgrade:staging	promote	install:production
<a href="#">bind-dev:el8:build</a>	<a href="#">bind-dev:el8:push</a>	<a href="#">bind-dev:el8:install:staging</a>	<a href="#">bind-dev:el8:update:staging</a>	<a href="#">bind-esv-to-bind-dev:el8:upgrade:staging</a>	<a href="#">bind-dev:el8:promote</a>	<a href="#">bind-dev:el8:install:production</a>
<a href="#">bind-dev:el9:build</a>	<a href="#">bind-dev:el9:push</a>	<a href="#">bind-dev:el9:install:staging</a>	<a href="#">bind-dev:el9:update:staging</a>	<a href="#">bind-esv-to-bind-dev:el9:upgrade:staging</a>	<a href="#">bind-dev:el9:promote</a>	<a href="#">bind-dev:el9:install:production</a>
<a href="#">bind-esv:el7:build</a>	<a href="#">bind-esv:el7:push</a>	<a href="#">bind-esv:el7:install:staging</a>	<a href="#">bind-esv:el7:update:staging</a>	<a href="#">bind-esv-to-bind:el7:upgrade:staging</a>	<a href="#">bind-esv:el7:promote</a>	<a href="#">bind-esv:el7:install:production</a>
<a href="#">bind-esv:el8:build</a>	<a href="#">bind-esv:el8:push</a>	<a href="#">bind-esv:el8:install:staging</a>	<a href="#">bind-esv:el8:update:staging</a>	<a href="#">bind-esv-to-bind:el8:upgrade:staging</a>	<a href="#">bind-esv:el8:promote</a>	<a href="#">bind-esv:el8:install:production</a>
<a href="#">bind-esv:el9:build</a>	<a href="#">bind-esv:el9:push</a>	<a href="#">bind-esv:el9:install:staging</a>	<a href="#">bind-esv:el9:update:staging</a>	<a href="#">bind-esv-to-bind:el9:upgrade:staging</a>	<a href="#">bind-esv:el9:promote</a>	<a href="#">bind-esv:el9:install:production</a>
<a href="#">bind:el7:build</a>	<a href="#">bind:el7:push</a>	<a href="#">bind:el7:install:staging</a>	<a href="#">bind:el7:update:staging</a>	<a href="#">bind-to-bind-dev:el8:upgrade:staging</a>	<a href="#">bind:el7:promote</a>	<a href="#">bind:el7:install:production</a>
<a href="#">bind:el8:build</a>	<a href="#">bind:el8:push</a>	<a href="#">bind:el8:install:staging</a>	<a href="#">bind:el8:update:staging</a>	<a href="#">bind-to-bind-dev:el9:upgrade:staging</a>	<a href="#">bind:el8:promote</a>	<a href="#">bind:el8:install:production</a>
<a href="#">bind:el9:build</a>	<a href="#">bind:el9:push</a>	<a href="#">bind:el9:install:staging</a>	<a href="#">bind:el9:update:staging</a>		<a href="#">bind:el9:promote</a>	<a href="#">bind:el9:install:production</a>

# BIND 9 vs. survey – packages



# BIND 9 team vs. survey

	Team priority	Survey priority
CVE frequency	#1	# 8
CI & automated tests	#1	# 11
code reviews & standards	#1	# 13

# Discussion

# Thank you!

- Main website: <https://www.isc.org>
- Software downloads:  
<https://www.isc.org/download> or  
<https://downloads.isc.org>
- Presentations: <https://www.isc.org/presentations>
- Main GitLab: <https://gitlab.isc.org>

