

Analysis of the Swiss RSE Survey 2025

12th of October 2026

Contributors:

Uwe Schmitt, Scientific IT Services, ETH Zurich

Emanuel Schmid, Scientific IT Services, ETH Zurich

Stuart Dennis, Scientific IT, Eawag Dübendorf

Contact: uwe.schmitt@ethz.ch

This document is licensed under [CC BY-NC 4.0](#)

DOI 10.5281/zenodo.23315160

About this survey

The Swiss RSE Survey 2025 collected responses from 192 participants from all over Switzerland between September and December 2025, mapping the professional landscape of Research Software Engineers (RSEs) in Switzerland: their backgrounds, working conditions, skills, and career trajectories. Participants were recruited using multiple public communication channels, such as the rse.swiss mailing list, the RSE related chat rooms and on LinkedIn.

The survey was conducted and analysed by the [RSE4ORD](#) project, funded by the ETH domain Open Research Data (ORD) programme. RSE4ORD builds a Research Software Engineering community in Switzerland to promote open science.

How this report is organised. The report opens with three insights drawn from the survey:

- **Insight: Gender Imbalance and Career Satisfaction**, who is in the community, where the satisfaction difference appears, and what improvements the literature suggests.
- **Insight: A Learning Path for Junior RSEs**, what the survey shows about how advanced skills are distributed over the years of experience, and what this implies for early-career RSEs.
- **Insight: Dedicated RSE Teams**, what respondents in dedicated or partially dedicated teams report, compared with the rest.

The sections that follow describe the survey population and the main themes: identity, satisfaction, skills, learning demand, and open source and FAIR practices. The report closes with the detailed junior-versus-senior skills analysis behind the learning-path insight.

Disclaimer: This report was produced with AI assistance for data analysis, interpretation and writing. The survey is limited to RSEs who are aware of the RSE community and submitted their data; awareness varies across organizations and scientific disciplines, so findings should not be generalised to all Swiss RSEs. Some response categories were pooled to improve group sizes.

Contents

About this survey	1
Insight: Gender Imbalance and Career Satisfaction	2
Insight: A Learning Path for Junior RSEs	3
Insight: Dedicated RSE Teams	4
Overview: Survey population	5
RSE identity and community needs	7
Open source and FAIR adoption	8
Satisfaction scores	9
Satisfaction with future perspectives breakdowns	10
Satisfaction differences between RSEs working in dedicated RSE teams vs the rest	11
Skills distribution	12
Differences in skills distribution between RSE teams and the rest	13
Learning demand analysis	14
Skills analysis: junior vs senior developers	15
What Makes an RSE in the ETH Domain?	17

Insight: Gender Imbalance and Career Satisfaction

The survey workforce is 75% men, 20% women, with 1.6% non-binary and 3% who preferred not to say. Women are less likely to hold permanent contracts (26% vs 44% of men). The non-binary group (n=3) and those who preferred not to say (n=6) are too small for separate analysis; their figures are reported but not interpreted.

Women and men report nearly identical satisfaction with job content (average score 4.15 for women, 4.26 for men) and pay (3.69 vs 3.67); organisation (3.44 vs 3.73) and recognition (3.49 vs 3.77) are somewhat lower for women. **The widest gap in this sample is for satisfaction with future prospects: 2.64 for women vs 3.26 for men.** The non-binary (n=3) and “prefer not to say” (n=6) groups are too small for further analysis and are therefore reported but not interpreted.

As the survey did not ask about hiring and promotion practices, the data does not allow us to conduct further analysis and provide tangible recommendations. However, the latest research suggests the following actions:

- **Train hiring committees.** Bias-awareness training for faculty search committees has been followed by higher hiring of women in STEMM departments. At one institution, the training also changed how faculty described the department climate ([Carnes et al., 2015](#); [Devine et al., 2017](#)).
- **Use criteria-first hiring.** A systematic review of 27 controlled studies found that bias against women in evaluation disappeared when the criteria for the job were fixed before applicants were reviewed, and when women made up at least a quarter of the applicant pool ([Isaac et al., 2009](#)).
- **Protect retention through contracts.** Women in this survey are far less likely to hold permanent contracts (26% vs 44% of men). This matters because parenthood carries a measurable penalty: in a field study of job applications, mothers were rated as less competent and offered lower starting salaries than equally qualified childless women ([Correll et al., 2007](#)).
- **Make senior women visible.** Role models change how achievable a career looks: women who saw senior women in their field rated their own prospects higher ([Marx & Roman, 2002](#); [Cheryan et al., 2011](#)), and a meta-analysis of lab and field studies links in-group role models to greater interest and performance ([Lawner et al., 2019](#)).

Insight: A Learning Path for Junior RSEs

The survey shows two different patterns when junior RSEs (fewer than 5 years of experience) are compared with senior RSEs (more than 15 years). Four development practices

- requirements gathering,
- design patterns,
- software architecture
- and code review

show an **exposure gap**: 30–42% of junior RSEs report no knowledge of them. Senior RSEs report such gaps in the range 2-12%.

Most other practices show a **depth gap**: junior RSEs already report basic knowledge, and the difference from senior RSEs is in advanced proficiency.

AI is the exception in the opposite direction: junior RSEs report more advanced knowledge of AI than senior RSEs (32% vs 12%), the only practice in the survey where juniors lead.

Advanced proficiency rises with years of experience, which suggests it is developed through sustained practice rather than in short training formats. The largest gaps are in practices that are usually exercised with others (code review, pair programming, architecture and design decisions), and the survey's team comparison points in the same direction: RSEs in dedicated teams report higher advanced rates in exactly these areas.

For early-career RSEs this suggests two measures:

- introductions to the four exposure-gap skills, and
- opportunities to practise the other skills alongside more experienced colleagues (mentoring, pair programming and code review) rather than more courses.

This is consistent with knowledge-management research: tacit skills are primarily transferred through practice with others, not from manuals or courses ([Björnson & Dingsøyr, 2008](#)).

Figures and definitions: Skills analysis: junior vs senior developers; what juniors want to learn: Learning demand analysis.

Insight: Dedicated RSE Teams

The survey shows a consistent pattern: respondents in dedicated or partially dedicated teams report higher satisfaction with recognition and with how their work is organised, and higher proficiency in several engineering practices, than respondents outside such teams. These are observed differences between groups; the survey is observational and does not establish that team membership causes them.

Lack of institutional support for RSEs is a persistent issue documented across multiple countries ([Hettrick et al., 2014](#); [Carver et al., 2022](#)), and dedicated RSE groups have been proposed as a structural response ([Cohen et al., 2021](#)).

RSEs in dedicated teams report higher satisfaction with recognition and with how their work is organised, and the differences are larger than for content, pay, and future prospects. This pattern is consistent with research showing that feeling valued at work depends more on organisational context (team climate, job design, managerial support) than on pay alone ([Gagné & Deci, 2005](#); [Judge et al., 2010](#)).

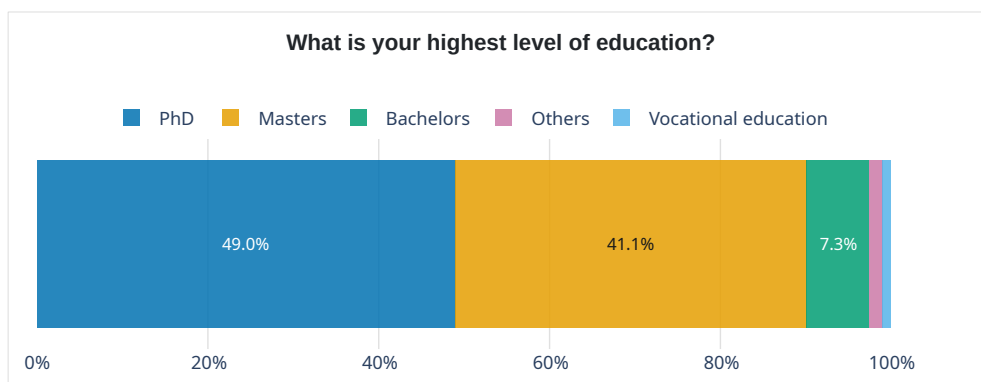
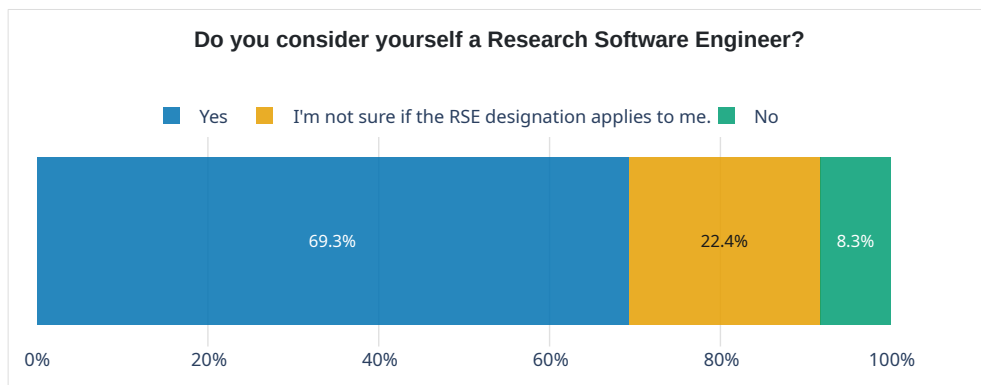
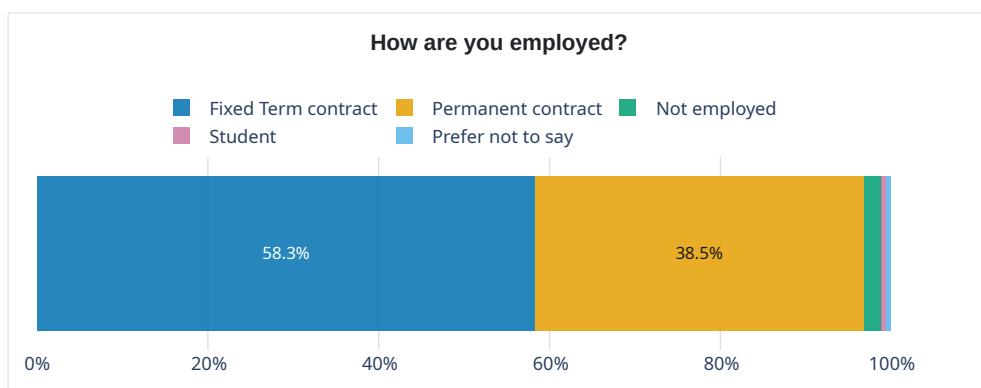
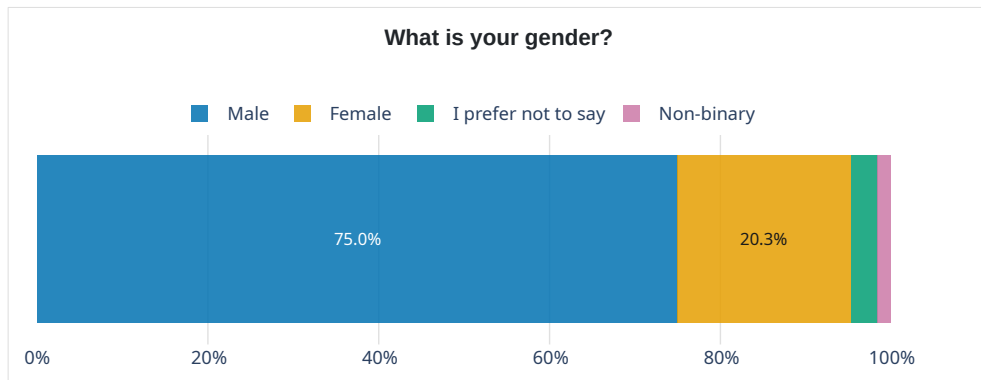
RSEs in dedicated teams also report higher proficiency in collaborative, quality-assurance and engineering practices, for example testing, code review, and back-end development. RSEs outside such teams are two to three times more likely to report no experience in code review, requirements gathering, back-end development, design patterns, and software architecture.

Code review and pair programming are collaborative practices. The literature links such skills to sustained participation in a community of practice ([Lave & Wenger, 1991](#)). Empirical work finds that pair programming improves quality and knowledge transfer ([Williams et al., 2000](#)); a meta-analysis of controlled experiments finds a small positive effect on quality, with more effort required ([Hannay et al., 2009](#)).

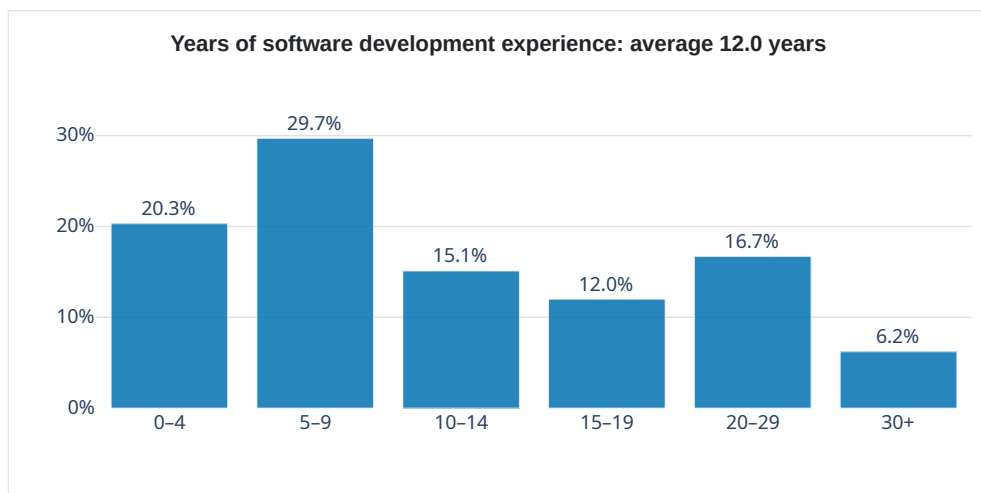
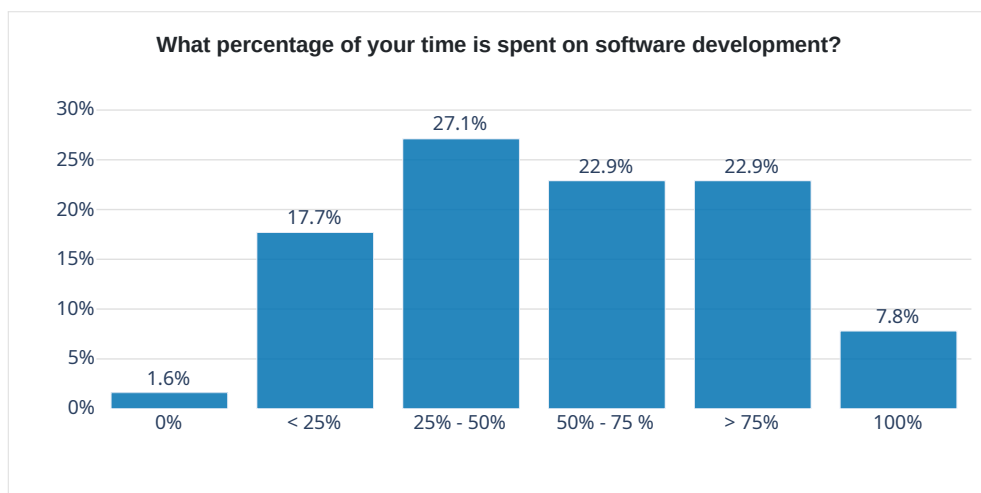
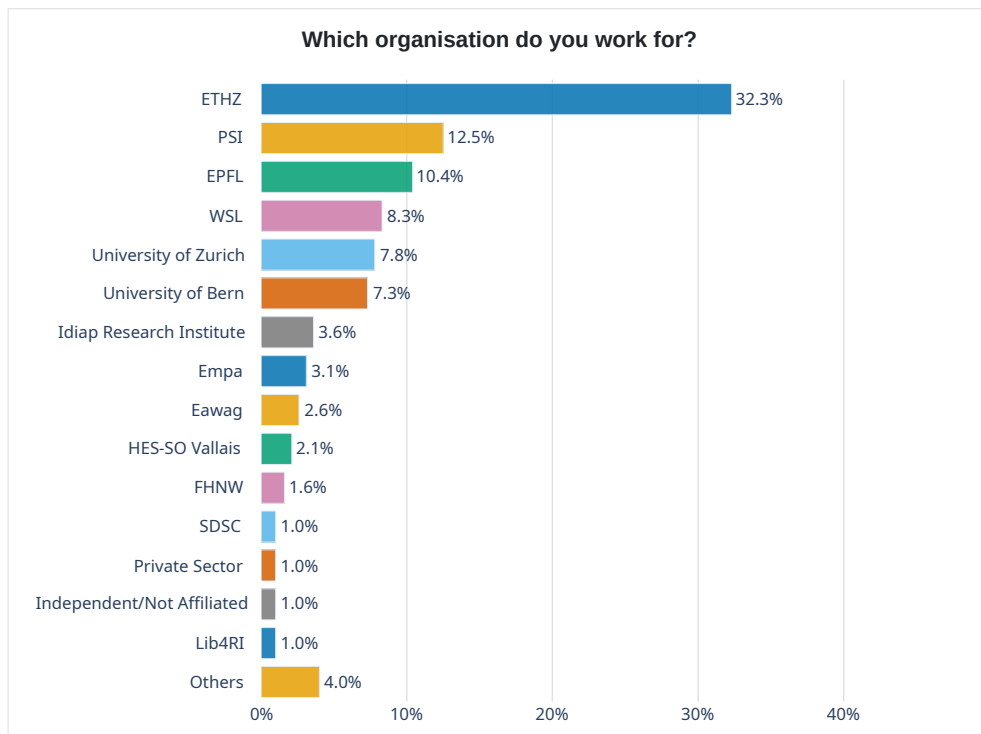
One might think that dedicated teams simply contain more experienced RSEs, and that this explains the differences we observed. But in this sample the groups are similar: median experience is 10 years in dedicated teams versus 8 years elsewhere, and the share reporting more than 15 years is nearly identical in both groups (27.3% of dedicated-team respondents versus 26.8% of the rest). The differences in reported proficiency remain within the 5–15 and 15+ year bands.

Dedicated-team respondents are also more likely to hold permanent contracts (44% vs 30% elsewhere). Contract type does not explain the proficiency differences either: they appear within both permanent and temporary/uncertain contracts, and for testing, code review and version control they are larger in the temporary employed group.

Overview: Survey population



Overview: Survey population (continued)

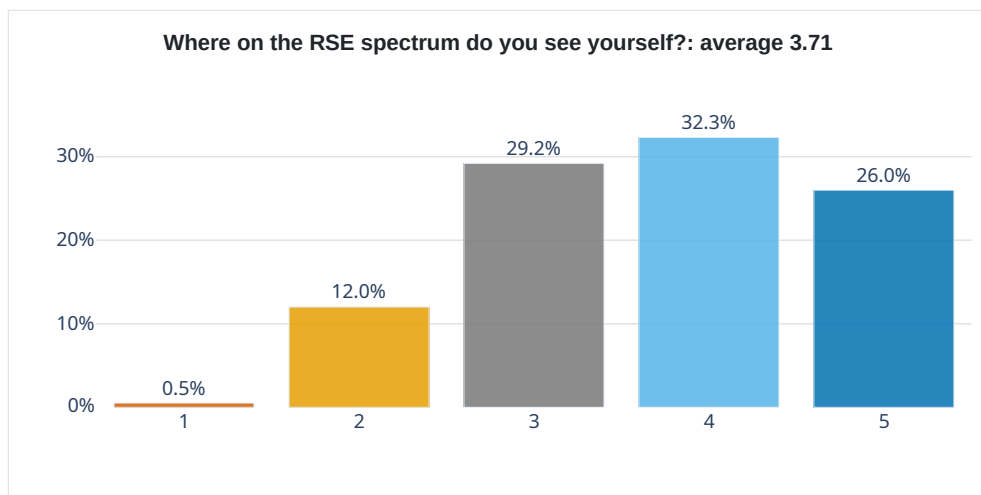


RSE identity and community needs

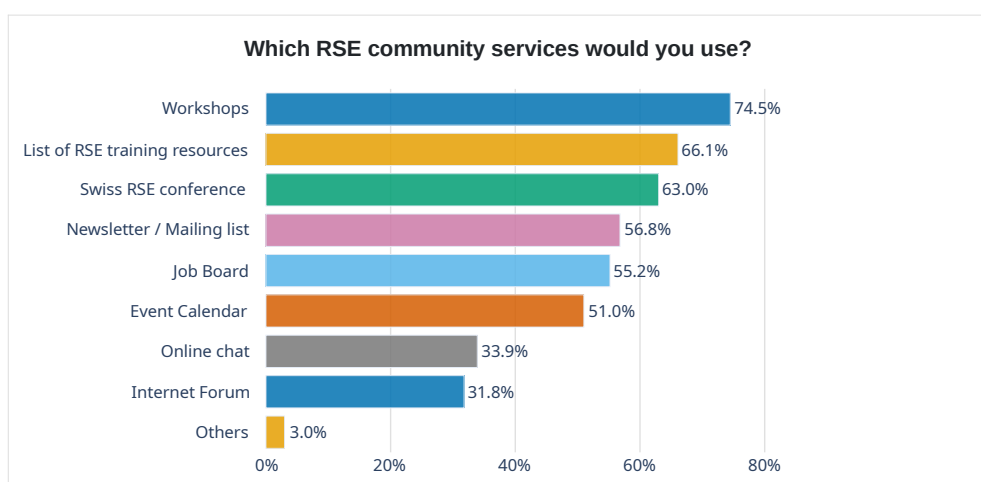
The roles of RSEs vary depending on the institutional context they work in. At one end of the spectrum, RSE roles may look similar to a traditional research role. At the other extreme, they resemble that of a software engineer in industry. We asked for a self-assessment on a scale from 1 to 5, where

- 1 corresponds to “pure researcher” and
- 5 to “pure software engineer”.

So the survey participants are rarely working in a very research focussed role, which also indicates a skew in the awareness of the RSE survey and the RSE4ORD activities.

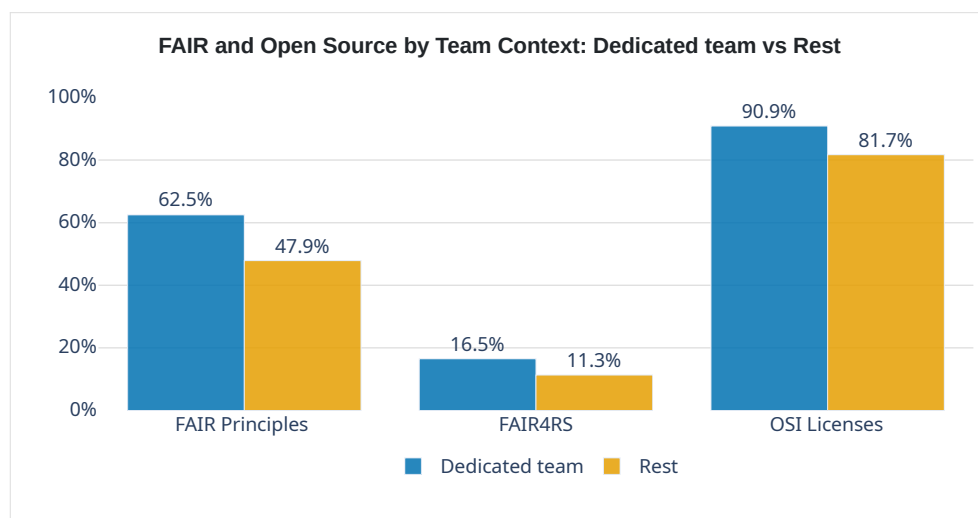


The respondents showed a high interest in learning related services, but also in a yearly Swiss RSE conference, newsletters and a job board.

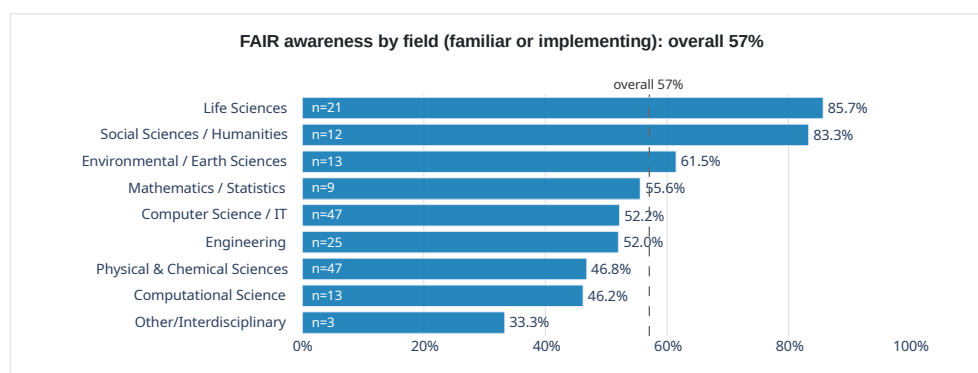


Open source and FAIR adoption

- **Open licensing is near-universal:** 87.5% of all respondents use OSI-approved licenses always or sometimes, 91% in dedicated teams vs 82% elsewhere.
- **FAIR awareness is moderate and uneven:** 57% overall (62% in dedicated teams vs 48% elsewhere), and 46% (Computational Science) to 86% (Life Sciences) by field.
- **FAIR4RS awareness is consistently low:** 15% overall (17% in dedicated teams vs 11% elsewhere), indicating it has yet to gain meaningful traction even among those familiar with broader FAIR principles.



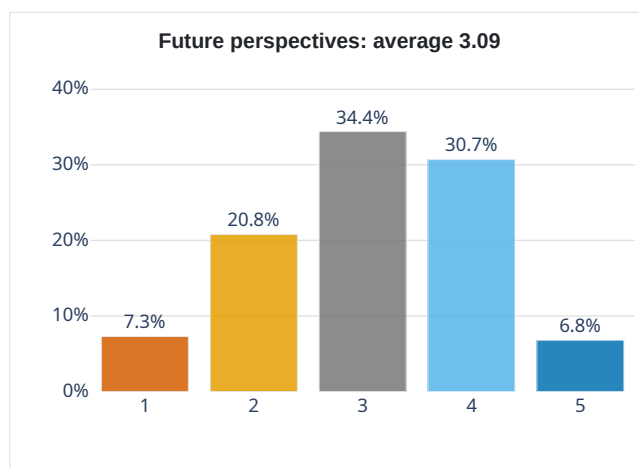
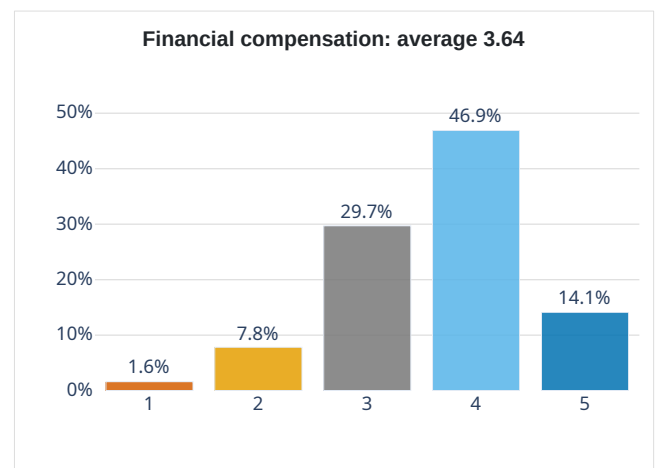
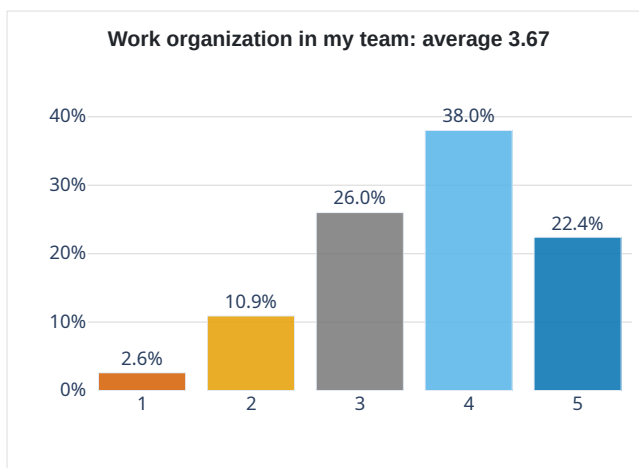
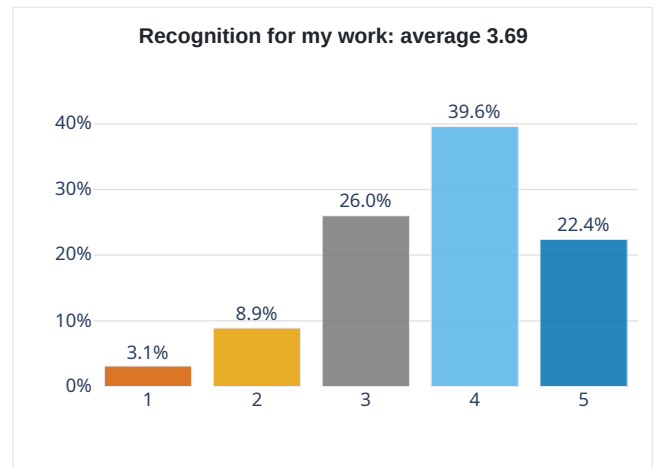
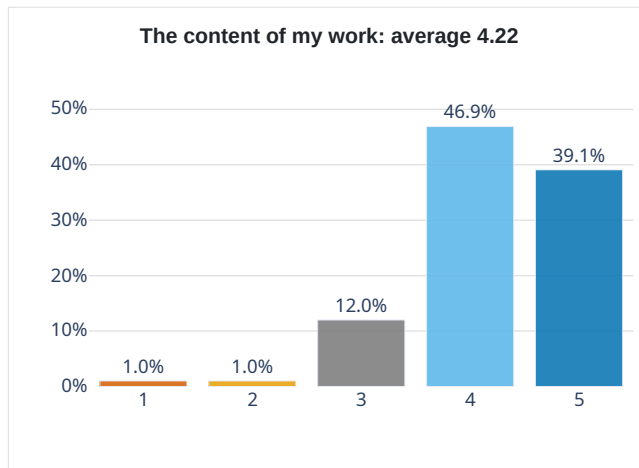
Familiarity with the FAIR principles varies between disciplines:



Satisfaction scores

Overall satisfaction with the content of work is highest and satisfaction with future perspectives ranks lowest.

Next come recognition for my work, work organization and financial compensation, with very similar average scores.



Satisfaction with future perspectives breakdowns

We analysed the scores and found differences related to gender and employment types.

Male respondents reported higher satisfaction with future perspectives than women, non-binary RSEs or respondents who did not want to specify their gender. The groups “Non-binary” and “I prefer not to say” are comparatively small and thus hard to interpret.

RSEs working in RSE teams or teams partially dedicated to RSE work also report a more positive outlook on their future perspectives.

RSEs on permanent contracts are also (not surprisingly) more satisfied with their future perspectives.



Satisfaction differences between RSEs working in dedicated RSE teams vs the rest

Teams dedicated to RSE work score higher on average in several aspects: the largest differences are in work recognition, work organisation and future perspectives. Differences in work content and pay are much smaller, suggesting the team effect is mainly about recognition and how work is organised rather than the nature of the work itself.



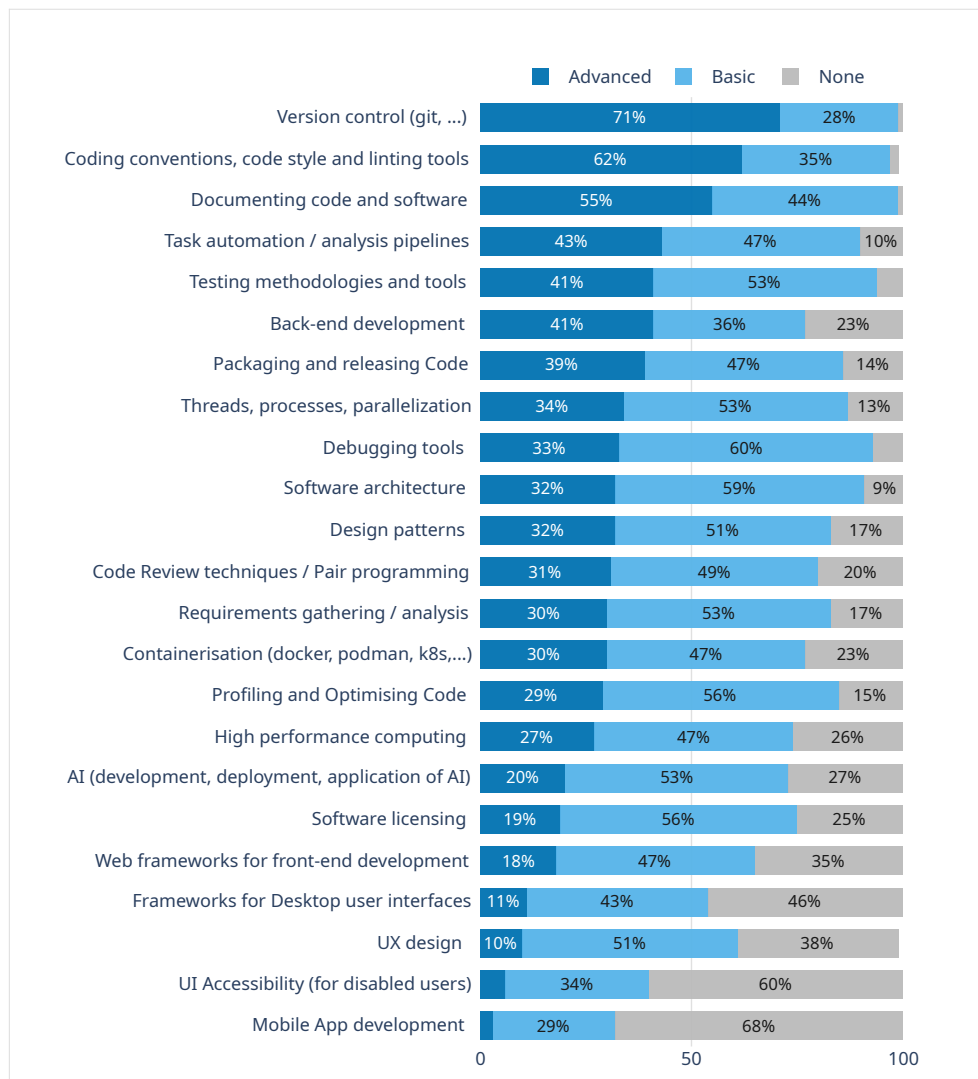
Skills distribution

Skills which are commonly known (almost no one reported no knowledge of these skills) among respondents are:

- Version control
- Coding conventions, code style and linting tools
- Documenting code and software

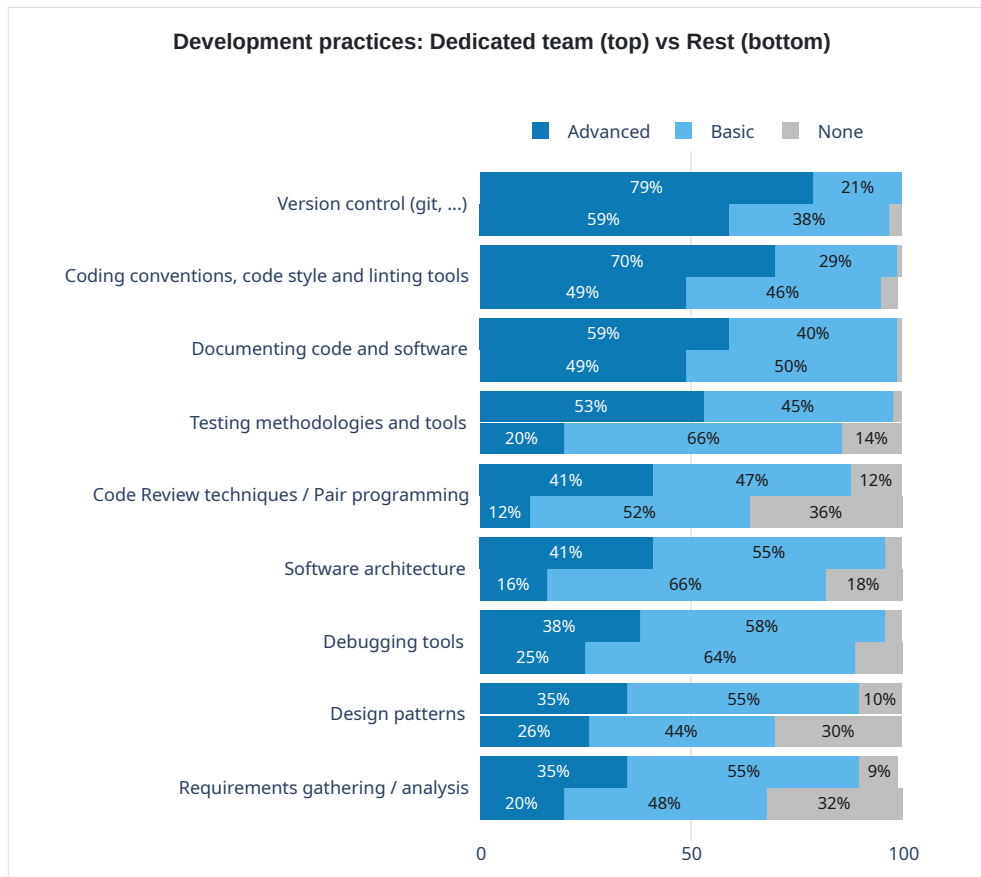
In contrast, skills related to user interface development show the lowest distribution among the survey participants. These are:

- Web frameworks
- Frameworks for Desktop user interfaces
- UX design
- UI Accessibility
- Mobile App development



Differences in skills distribution between RSE teams and the rest

The largest gaps in advanced knowledge are in testing methodologies and tools, code review / pair programming, and software architecture.



Learning demand analysis

Learning demand is broad and fairly evenly spread: no single skill dominates, and demand declines gradually across the ranking rather than falling away sharply.

A small top group leads, with around 38–40% of respondents wanting to learn each of software architecture, profiling and optimising code, and containerisation.

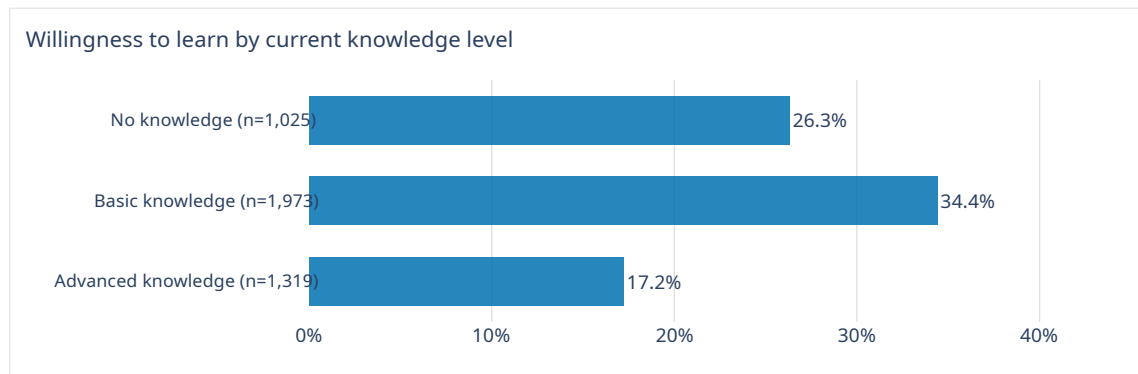
A wide middle band follows between roughly 29% and 35%, covering testing methodologies and tools, code review and pair programming, design patterns, parallelisation, debugging, AI, high-performance computing and requirements gathering.

The lowest-demand group sits around 18–25% and includes version control (18%), front-end and desktop UI frameworks, UI accessibility, mobile app development, documenting code and coding conventions. The spread between the most and least wanted skill is about 22 percentage points.



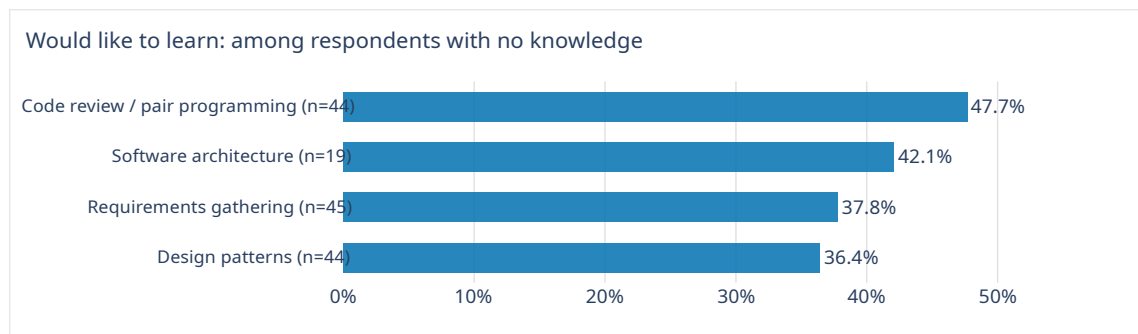
Demand on its own does not show whether a topic is a real gap, because respondents may simply want to deepen a skill they already use. The same question block also records how much knowledge respondents already have, which allows the two to be compared.

Across the 4,317 skill ratings collected, respondents report advanced knowledge for 31%, basic knowledge for 46% and no knowledge for 24%.



Willingness to learn is highest among respondents with basic knowledge of a skill (34%), lower among those with no knowledge (26%), and lowest among those with advanced knowledge (17%). The ordering suggests that motivation follows a foothold: people are more likely to want training in something they have already begun using than in something entirely unfamiliar.

The four exposure-gap skills are the exception. Among respondents with no knowledge of them, willingness to learn remains high:



At the other end, the skills that combine the least knowledge with the least demand are the UI-related ones

- desktop UI frameworks
- UI accessibility
- mobile app development
- and front-end web frameworks.

Those look like genuinely low-priority topics rather than hidden gaps.

This matches the learning-path insight, where the same four practices form the exposure gap for junior RSEs.

Skills analysis: junior vs senior developers

We compared junior RSEs (fewer than 5 years of software development experience) with senior RSEs (more than 15 years). The 5–15 year group is not shown in this comparison.

The survey's software-development-practices block covers nine items. Two measures are used: the share reporting advanced knowledge, and the share reporting at least basic knowledge.

Exposure gaps

Four practices show the largest differences in basic knowledge, with a substantial share of juniors reporting no knowledge at all:

Skill	Junior: basic or advanced skills	Senior: basic or advanced skills	Junior: no knowledge
Requirements gathering / analysis	58%	94%	42%
Design patterns	59%	92%	41%
Software architecture	70%	98%	30%
Code review / pair programming	64%	88%	36%

All four are also among the skills junior RSEs most often want to learn (see Learning demand analysis).

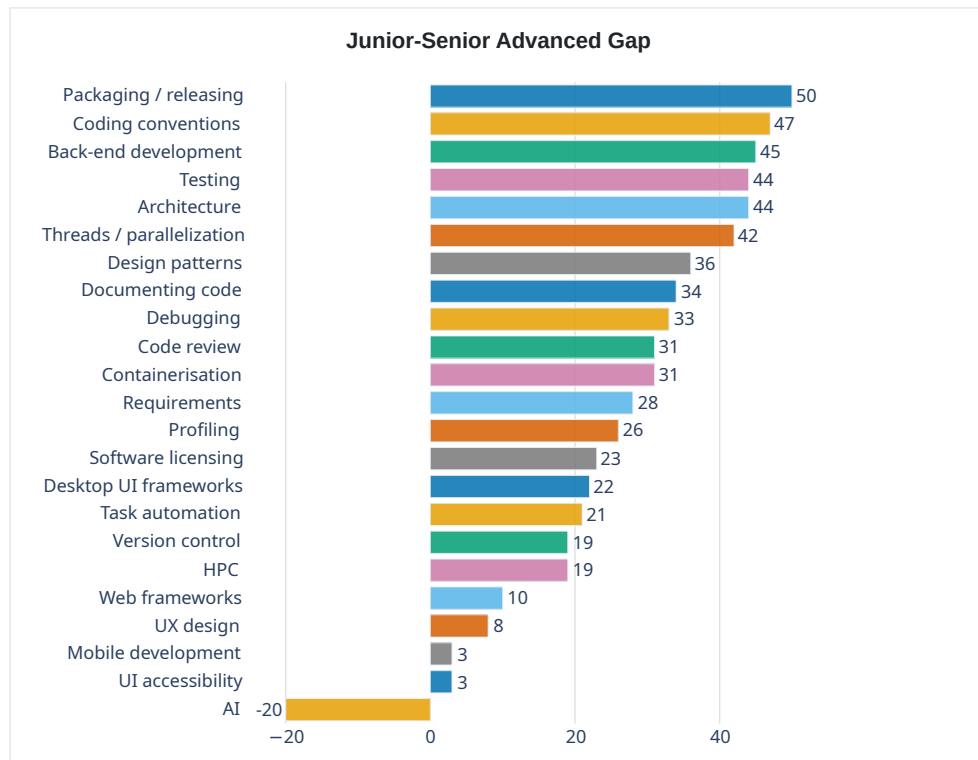
Depth comes with experience

For the other five practices, most junior RSEs already report basic or advanced knowledge; the difference from seniors is in advanced proficiency:

Skill	Junior: advanced	Senior: advanced	Junior: basic or advanced skills
Testing methodologies and tools	13%	58%	90%
Debugging tools	13%	46%	84%
Coding conventions, code style and linting	28%	76%	95%
Documenting code and software	29%	64%	100%
Version control (git, ...)	56%	75%	100%

Advanced proficiency rises with years of experience (testing: 13% to 58%); version control is the extreme case, with every junior reporting at least basic knowledge.

The chart below shows the senior–junior advanced knowledge gap in percentage points for all 23 skills.

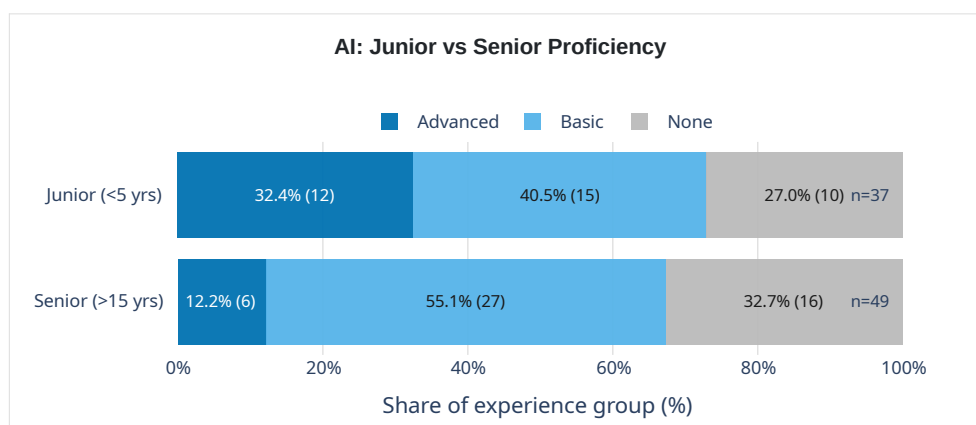


AI is the one skill where junior RSEs report more advanced knowledge than senior RSEs. The same direction appears across the three experience groups:

Experience group	Advanced AI knowledge	Wants to learn AI
Junior (<5 yrs)	32.4%	41.0%
Mid (5–15 yrs)	19.4%	28.7%
Senior (>15 yrs)	12.2%	28.8%

Advanced AI knowledge: respondents who rated the AI item (n=37 junior, 93 mid, 49 senior). Wants to learn AI: all respondents in each group (n=39, 101, 52).

Separately, 80% of respondents use AI tools for programming help; the chart below measures AI as a technical domain (development, deployment, application), not the use of AI assistants.



What Makes an RSE in the ETH Domain?

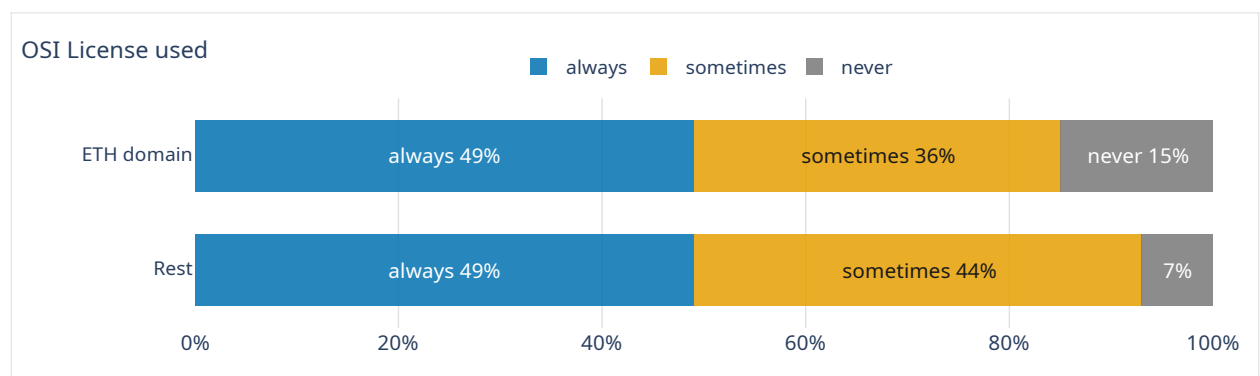
The ETH domain accounts for 137 of the 192 respondents (71.4%): ETH Zurich 62, PSI 24, EPFL 20, WSL 16, Empa 6, Eawag 5, SDSC 2 and Lib4RI 2. The remaining 55 work at other Swiss universities, research institutes and organisations.

Institution	Respondents	Share
ETH Zurich	62	32.3%
PSI	24	12.5%
EPFL	20	10.4%
WSL	16	8.3%
Empa	6	3.1%
Eawag	5	2.6%
SDSC	2	1.0%
Lib4RI	2	1.0%
ETH domain total	137	71.4%

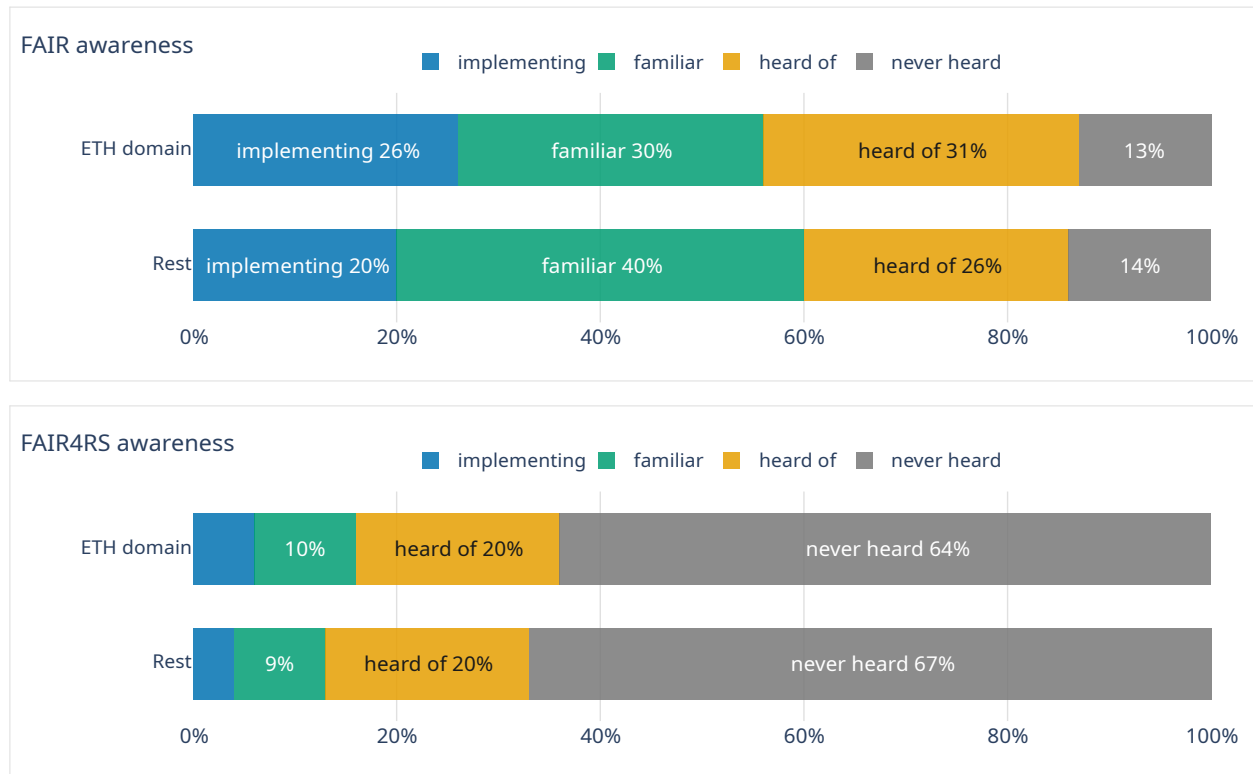
Compared with the rest of the sample, ETH-domain respondents are similar on role and team context. The clearest differences are disciplinary and educational: ETH-domain respondents are more likely to hold a PhD and to come from physical sciences or engineering, and less likely to come from computer science. Team context is similar in both groups.

Dimension	ETH domain	Rest
PhD	53.3%	38.2%
Physical & chemical sciences (field)	27.0%	18.2%
Engineering (field)	16.1%	5.5%
Computer science / IT (field)	19.7%	36.4%
In a dedicated or partially dedicated software team	63.5%	61.8%

ETH-domain respondents report using OSI-approved licences less often than the rest of the sample.



For the FAIR principles, a similar share of both groups has never heard of them (13% vs 14%), but the distribution across awareness levels differs: more ETH-domain respondents report implementing the principles (26% vs 20%) and fewer report being only familiar with them (30% vs 40%). FAIR4RS awareness is low in both groups:



The gap is not explained by the other dimensions measured in the survey. The groups are unequal in size (137 versus 55) and the survey has no population denominator per institution, so they are not equivalent samples. As this is an observational study, part of the gap may reflect who responded, and firm conclusions about institutional differences cannot be drawn.