

Exploring the BEAM European Community



Profile analysis of developers,
companies, and academia





Get the slides!



Maria José Gavilán
@Stritzinger



Icia Carro
@Stritzinger



EXPLORING THE EUROPEAN BEAM COMMUNITY



Do you ever feel overwhelmed by how scattered BEAM-related information is across different platforms? How easy is it to stay up to date and find the help you need?

Have you questioned how stable it feels to build your career around BEAM technologies? Is it a long-term, reliable choice for your growth? Are we working together as a community to drive a healthy sustainable future for the BEAM?

Our Drive

The BEAM has probably impacted your productivity and software quality, are there ways to improve even further? Are you aware of current successful case studies and experience reports across the wide community?

What is academia's role in the BEAM? Where is academia?

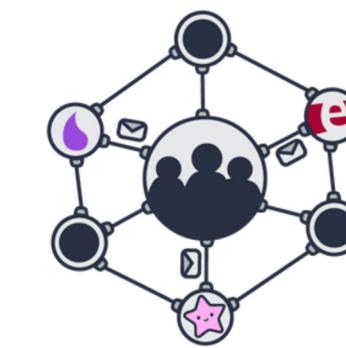
Are companies open about using BEAM languages?

Are businesses aware of the benefits of languages that run on BEAM?

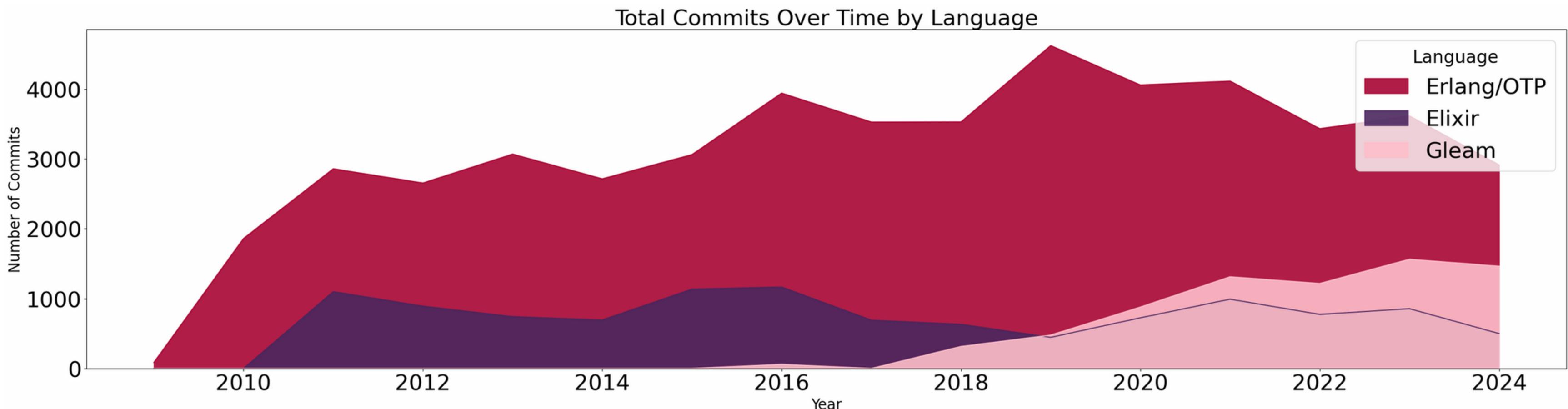
How happy are we with what we do?

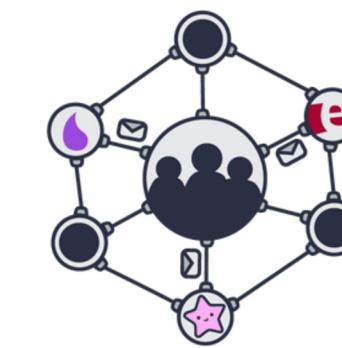
How accessible is our community? Could we do more to support newcomers and create a more welcoming environment? Can we improve the communication between the ecosystem?

A little context

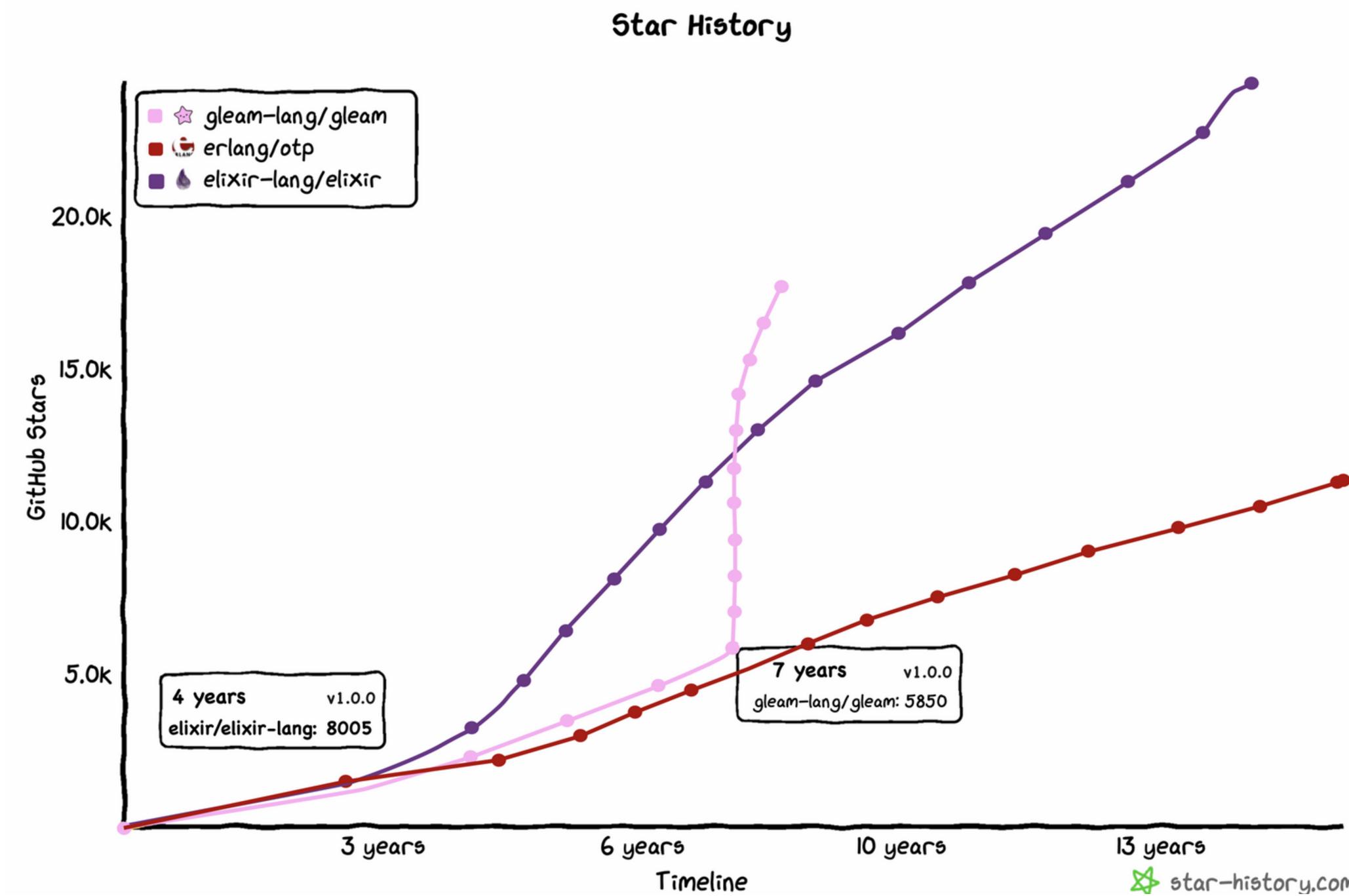


BEAM Languages Commits

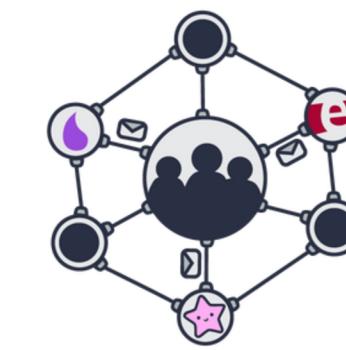




Erlang, Elixir, Gleam Start History

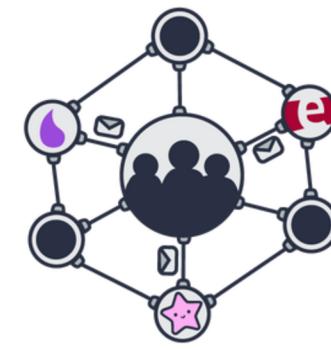


Reaching Out to Developers, Companies, and Academia: Gaining Insights for the Future



General Objective

Analyze the **adoption**, **diversity**, and **challenges** faced by developers, companies, and academia in Europe using **Erlang**, **Elixir**, and **GLEAM** languages on the BEAM, and assess their impact.

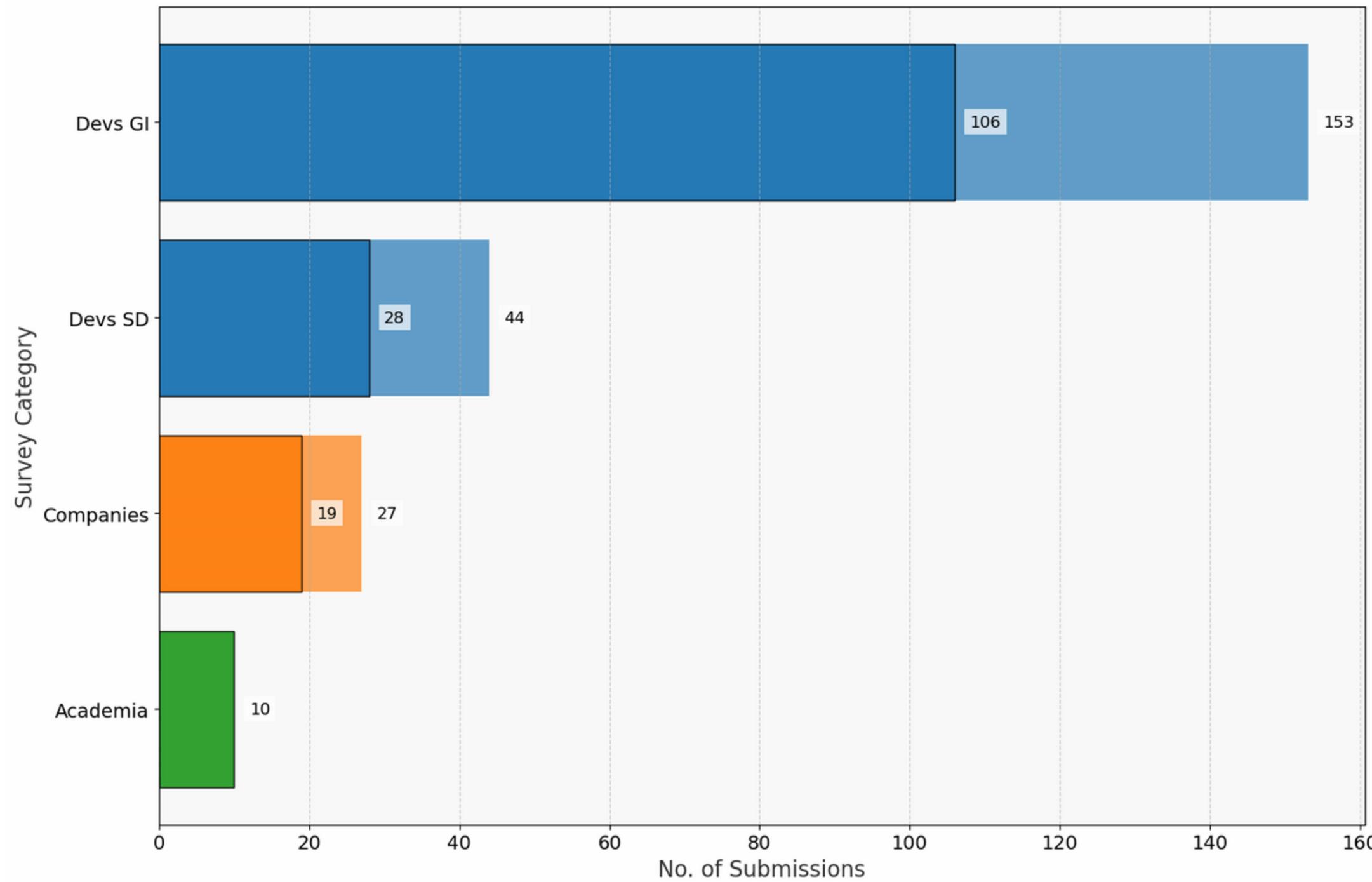
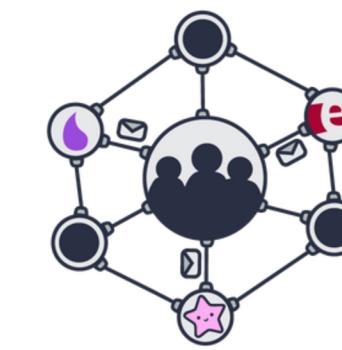


Specific Objectives

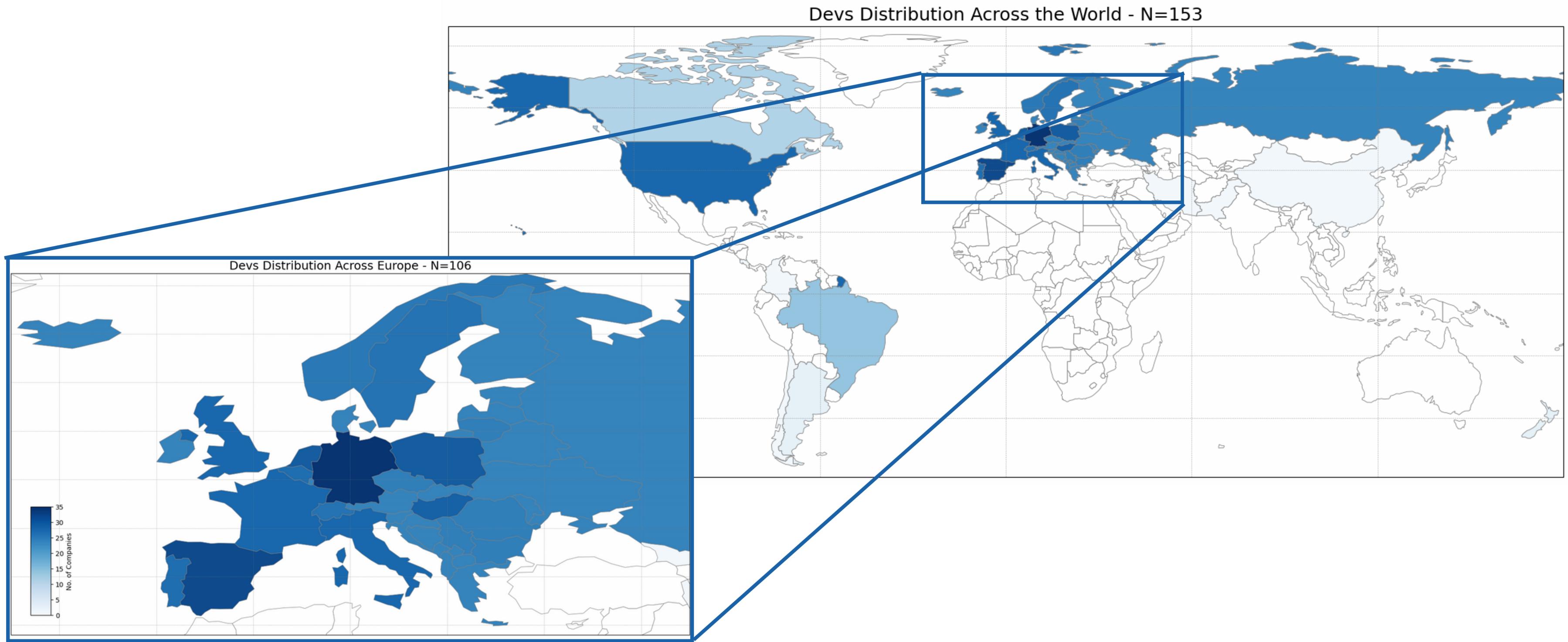
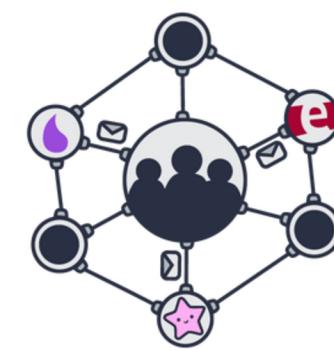
-  Profile BEAM community developers in Europe.
-  Assess the impact on productivity, software quality, and scalability.
- Explore motivations and challenges for companies and academia in adopting BEAM languages.
-  Evaluate diversity and inclusivity within the BEAM community.

A Sneak Peek at Our Findings

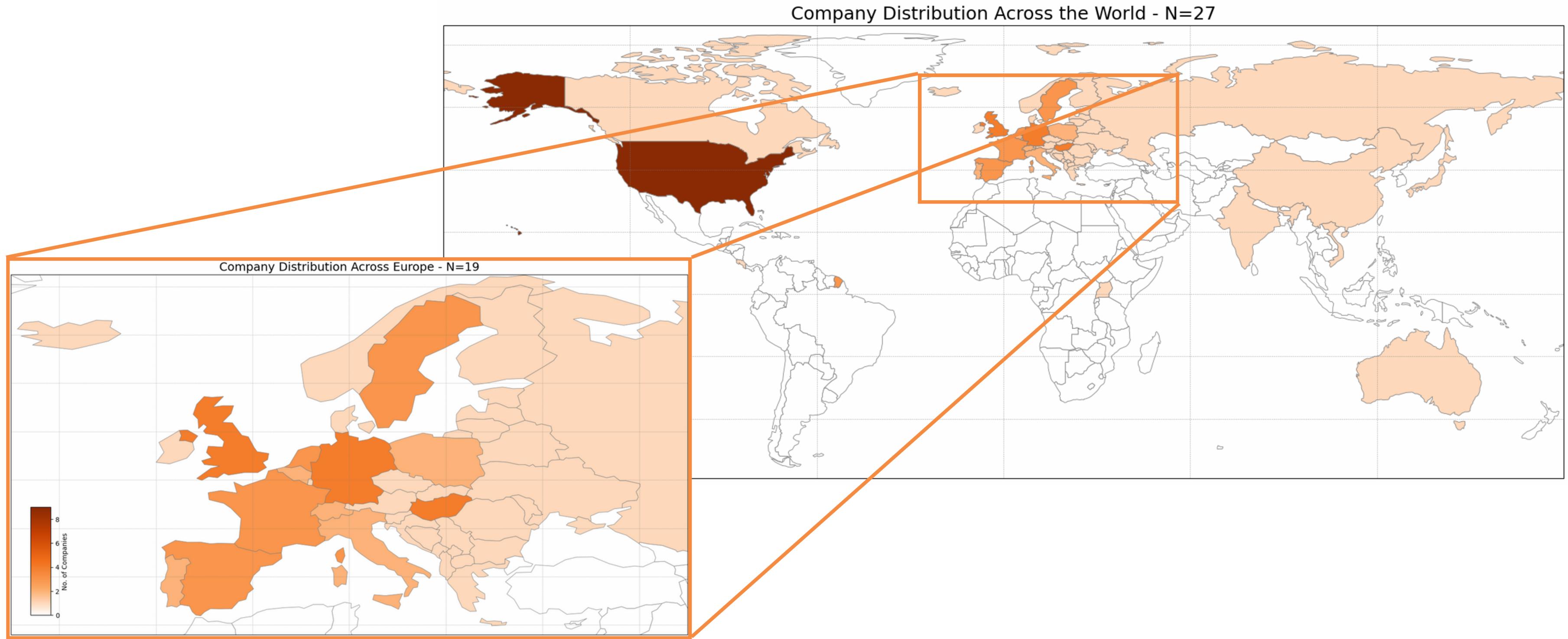
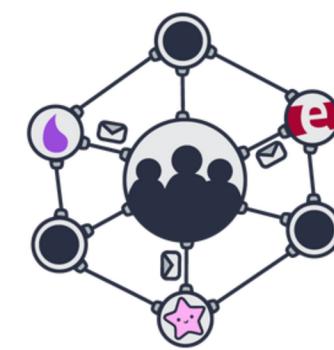
TOTAL RESPONSES



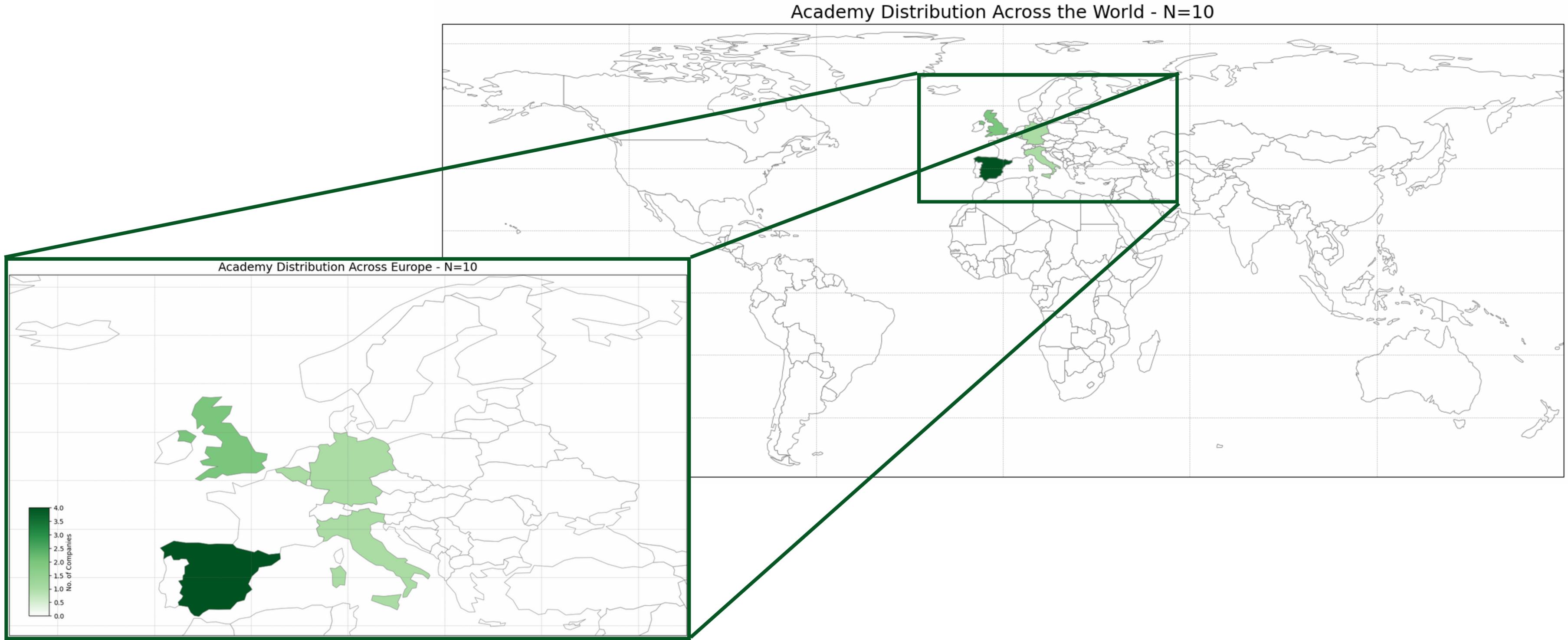
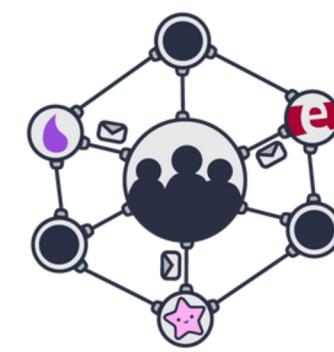
DEVELOPERS



COMPANIES

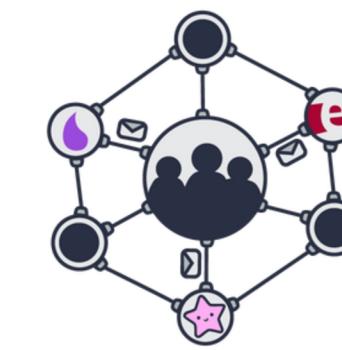


ACADEMIA

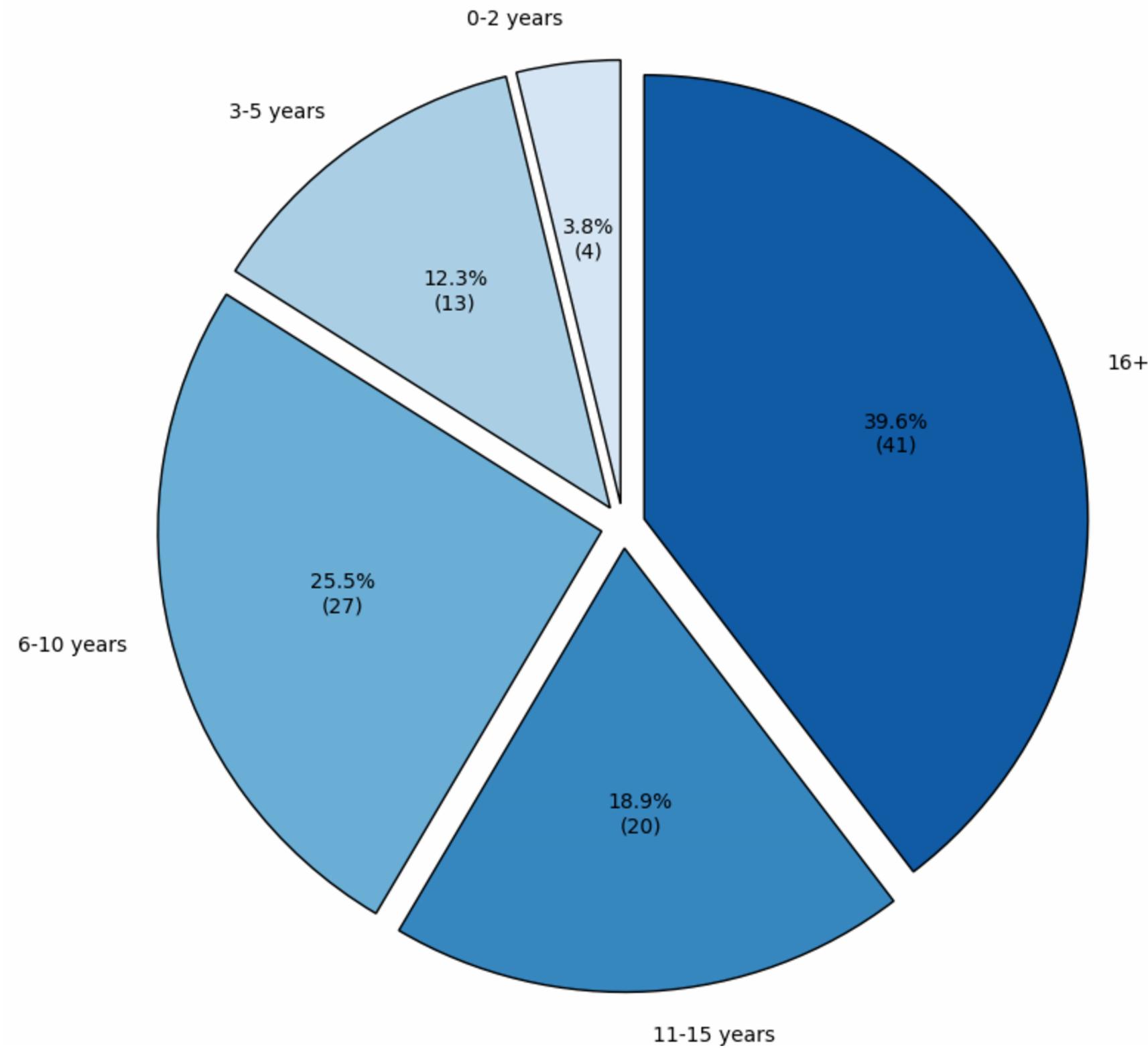


General Profile

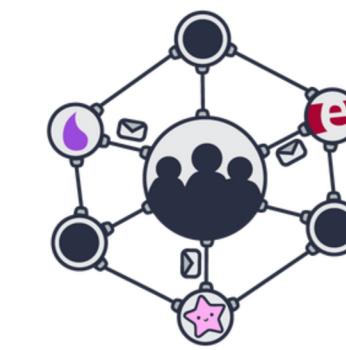
DEVELOPERS YEARS OF EXPERIENCE



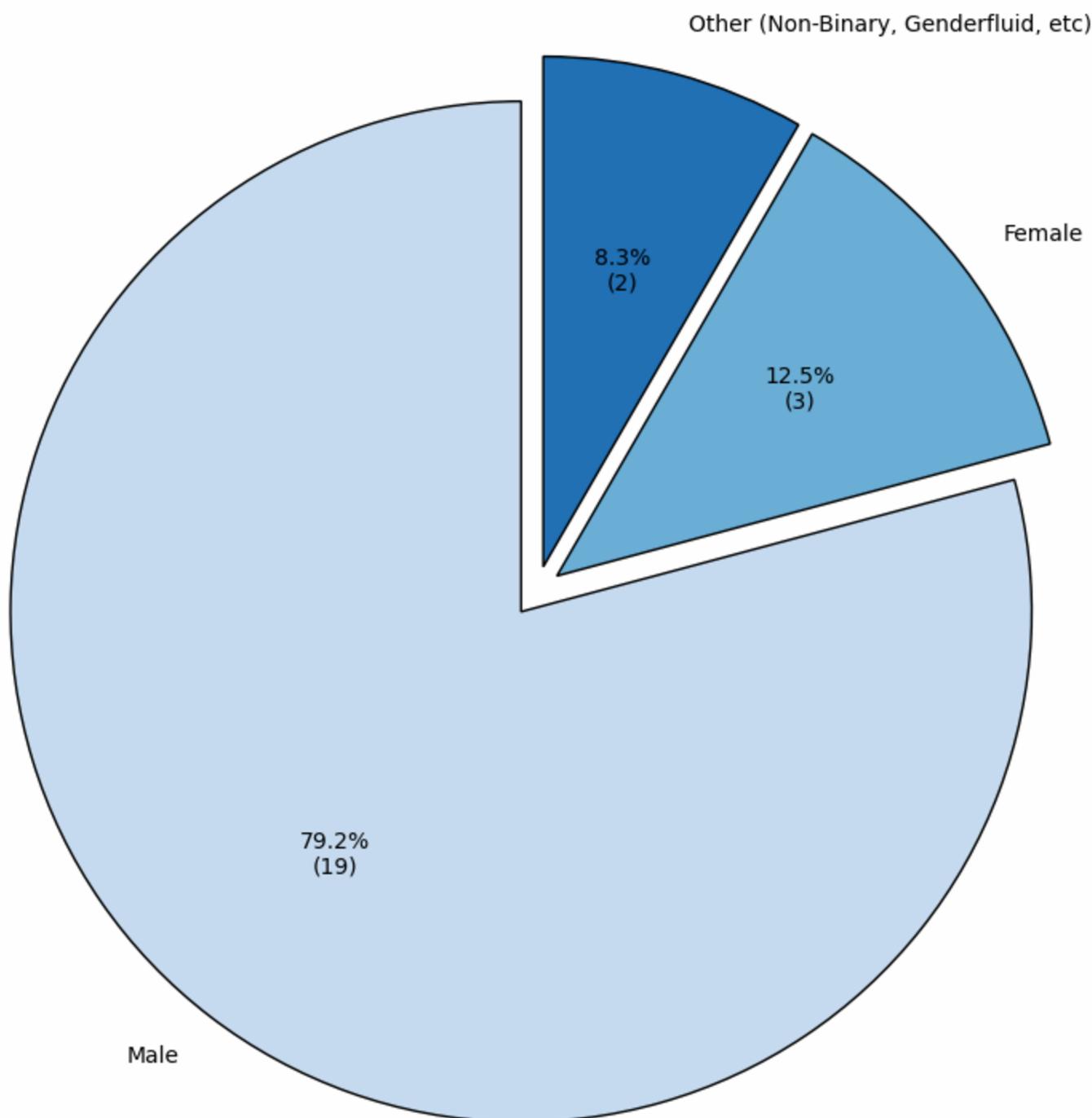
Distribution of Years of Experience (n=106)



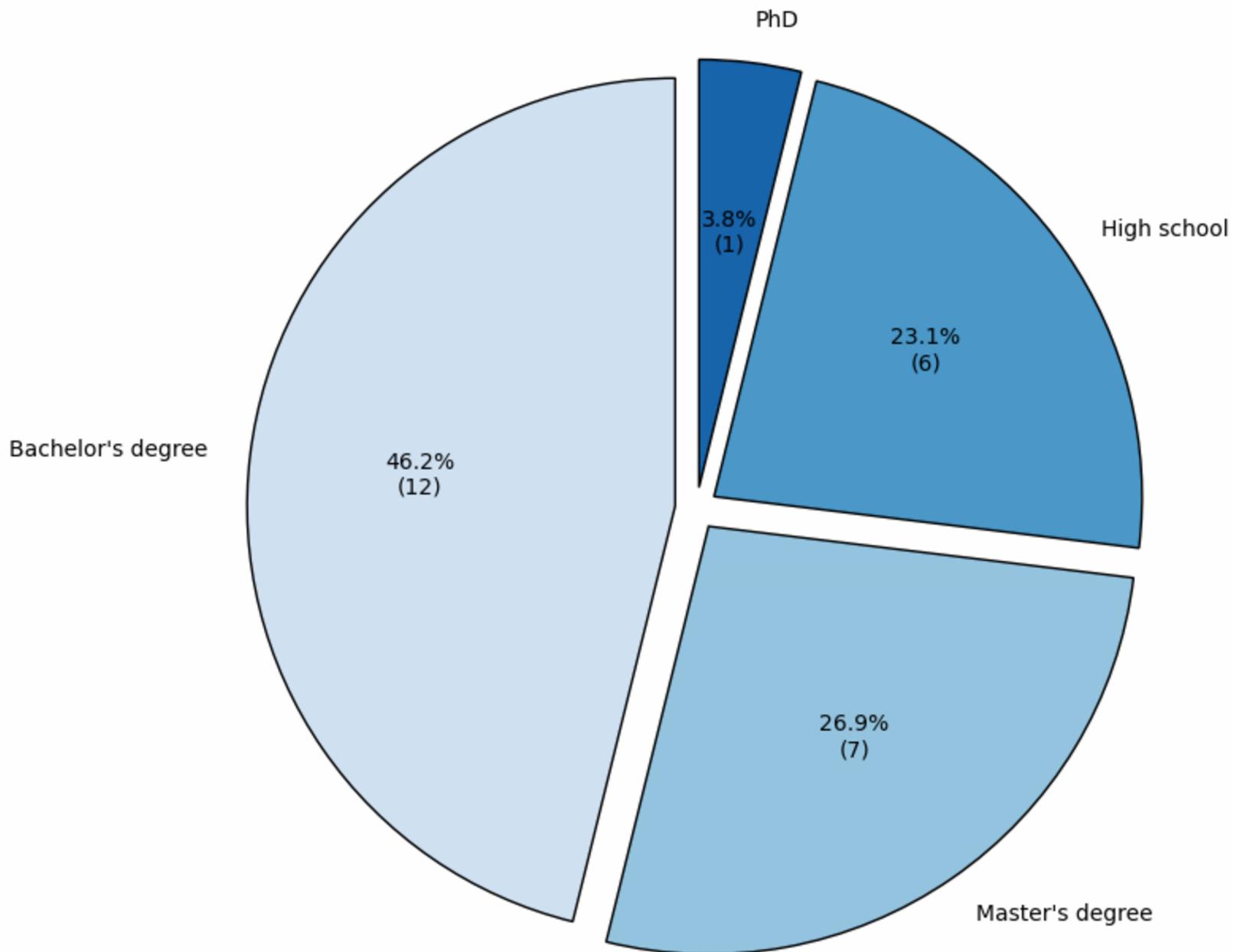
DISTRIBUTION OF GENDER REPRESENTATION & EDUCATIONAL BACKGROUND



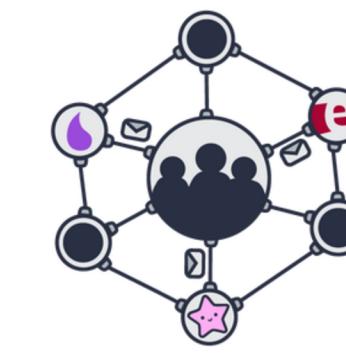
Distribution of Gender Identity (n=24)



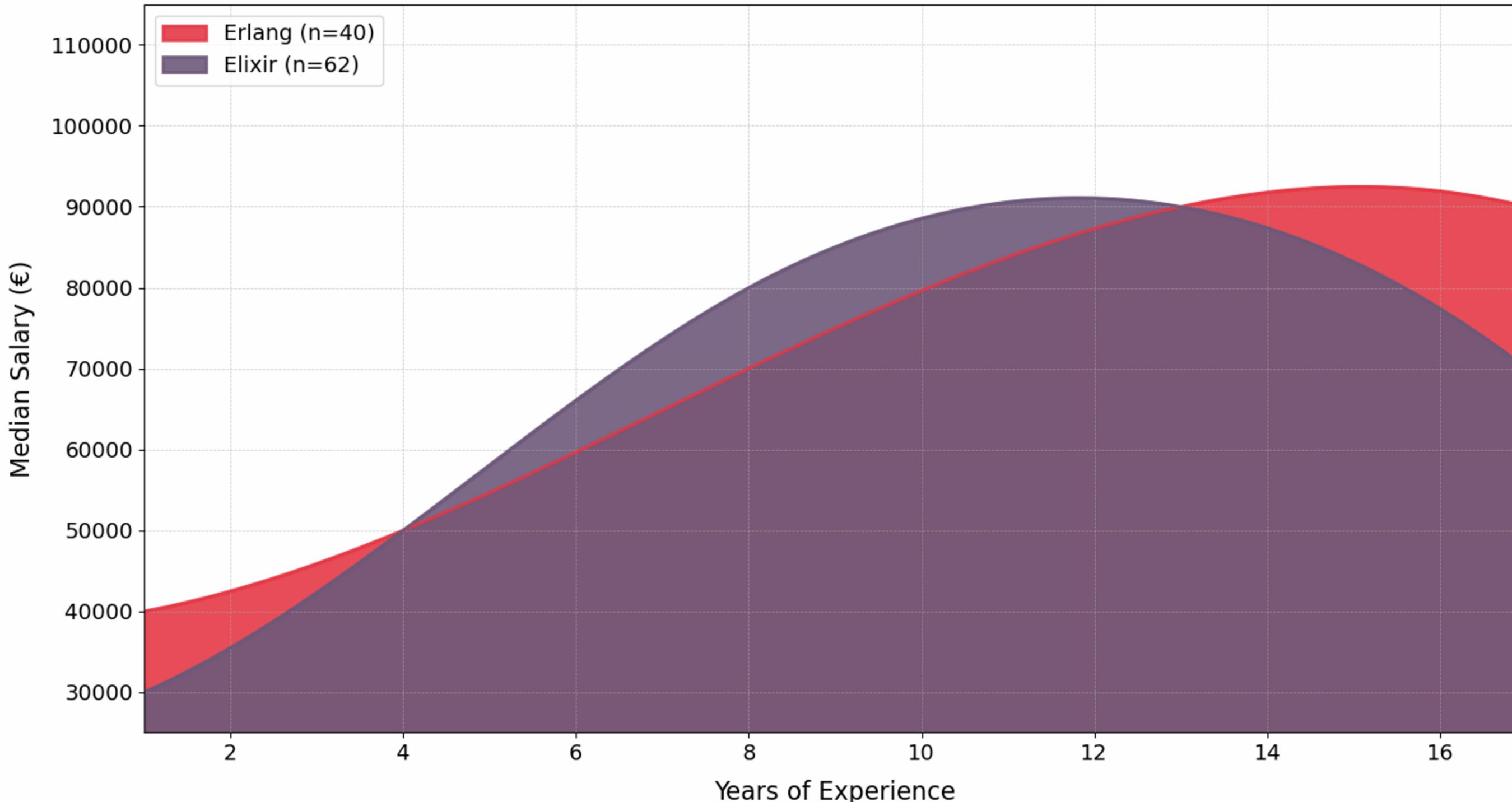
Distribution of Highest Level of Education (n=26)



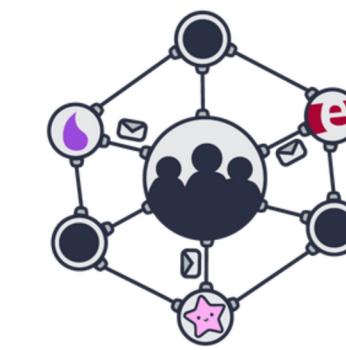
DEVELOPERS & COMPANIES: YEARS OF EXPERIENCE VS SALARY



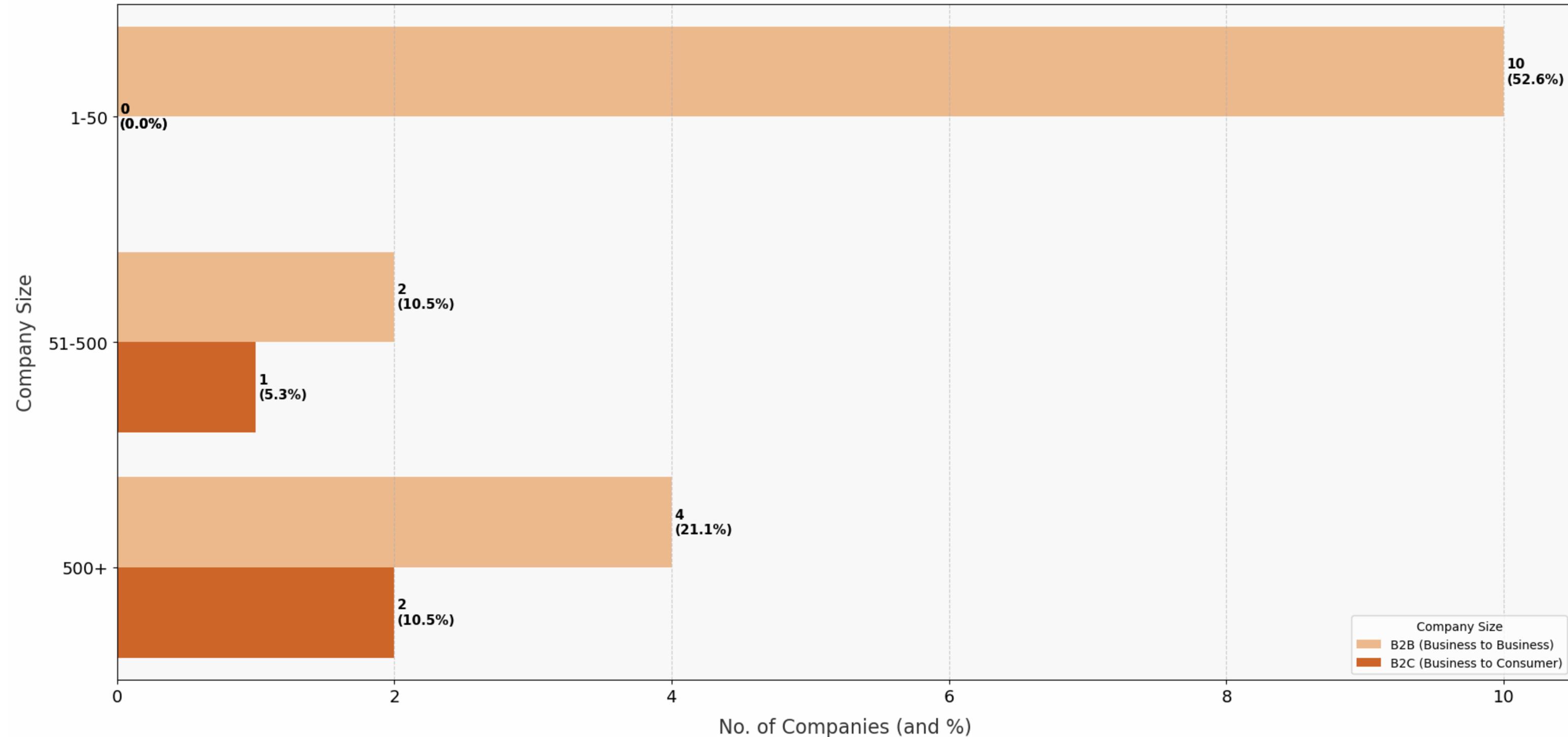
Median Salary by Years of Experience (Elixir & Erlang)



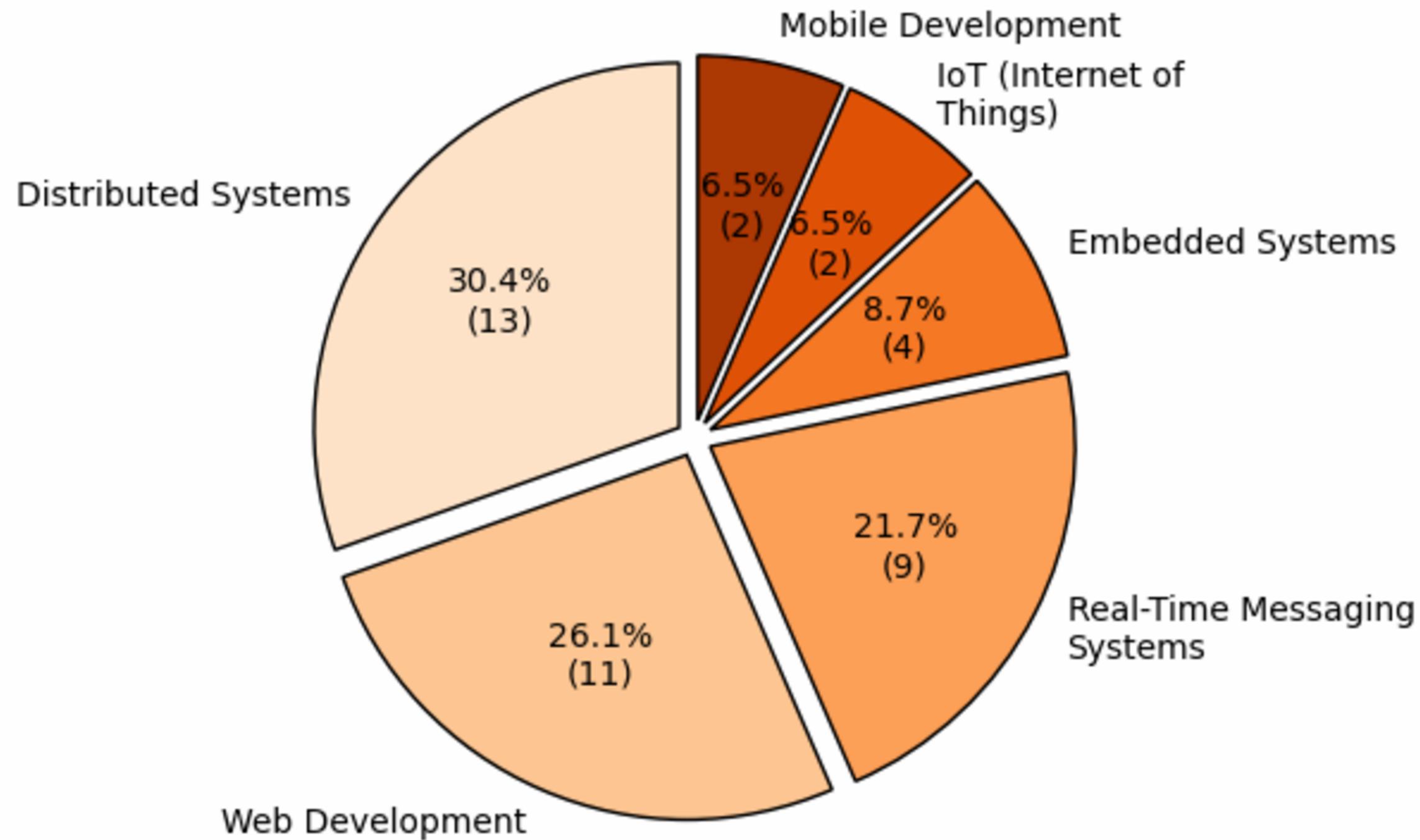
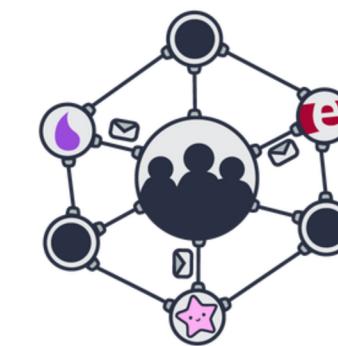
COMPANIES - BUSINESS MODEL



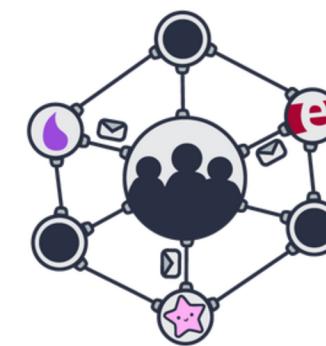
Company Sizes and Business Model, N=19



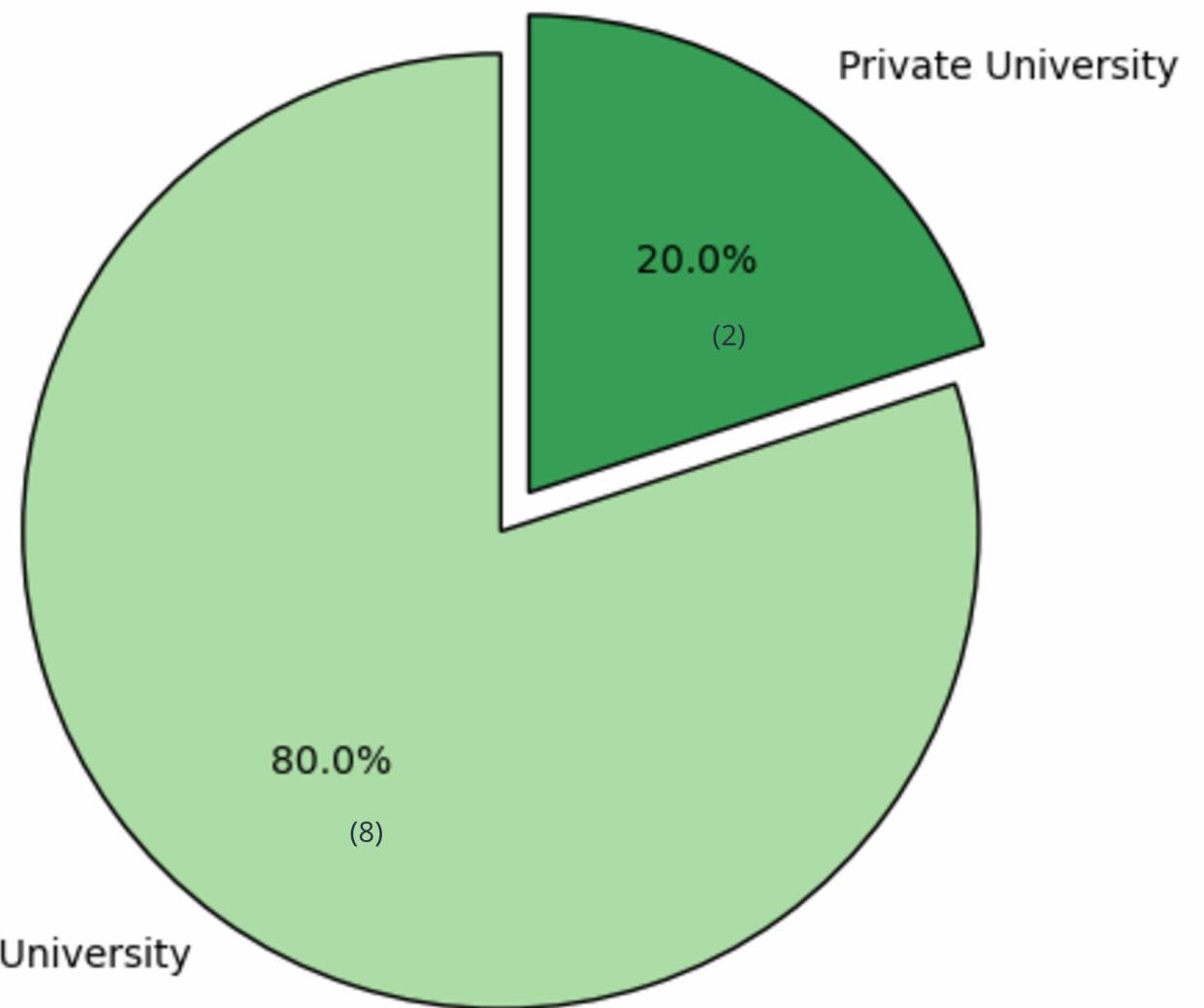
COMPANIES - TYPES OF PROJECTS



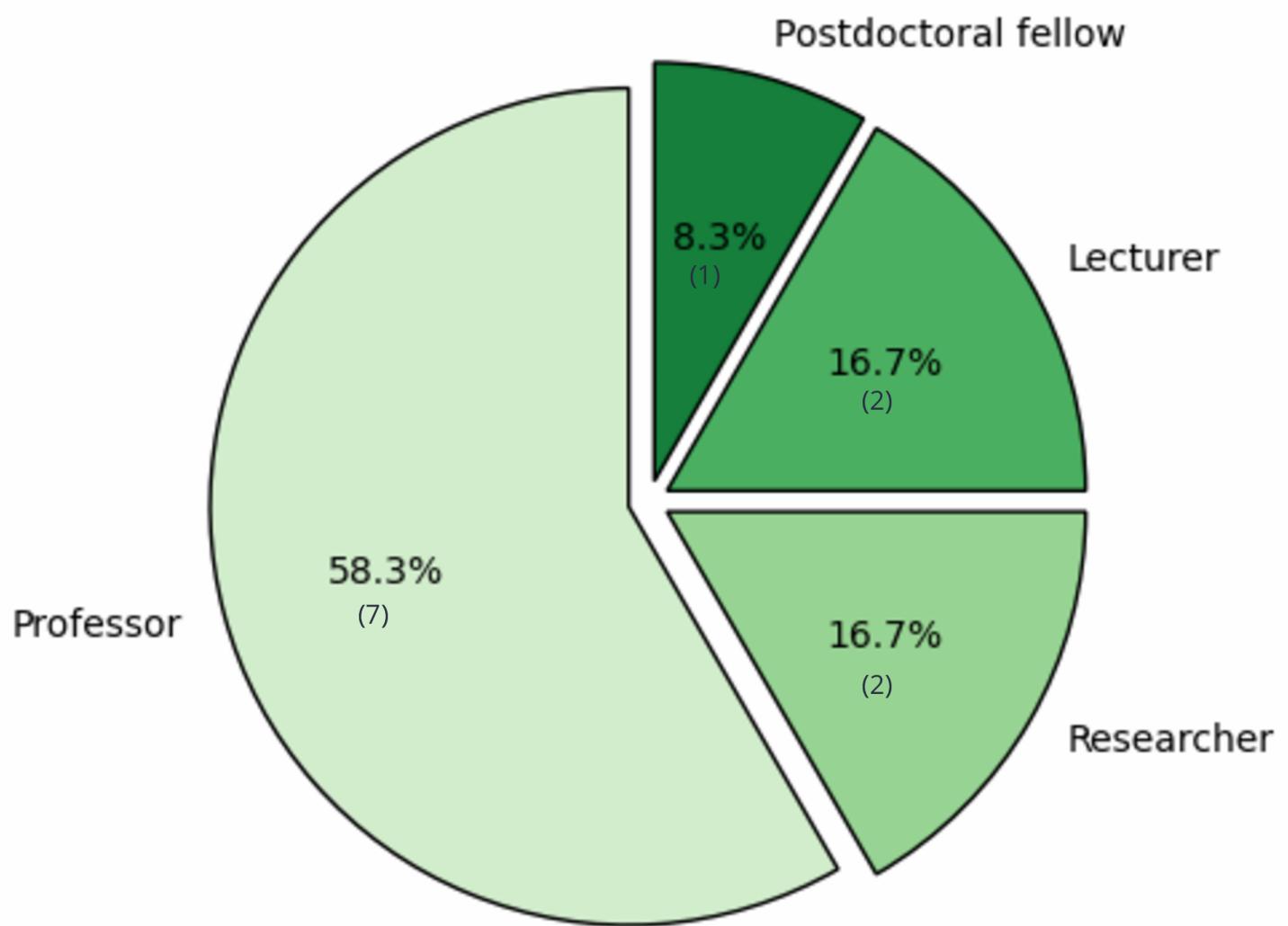
ACADEMIA



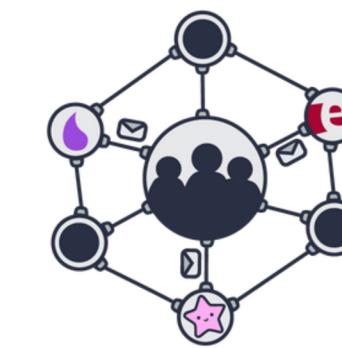
Distribution of Type of Institution (N=10)



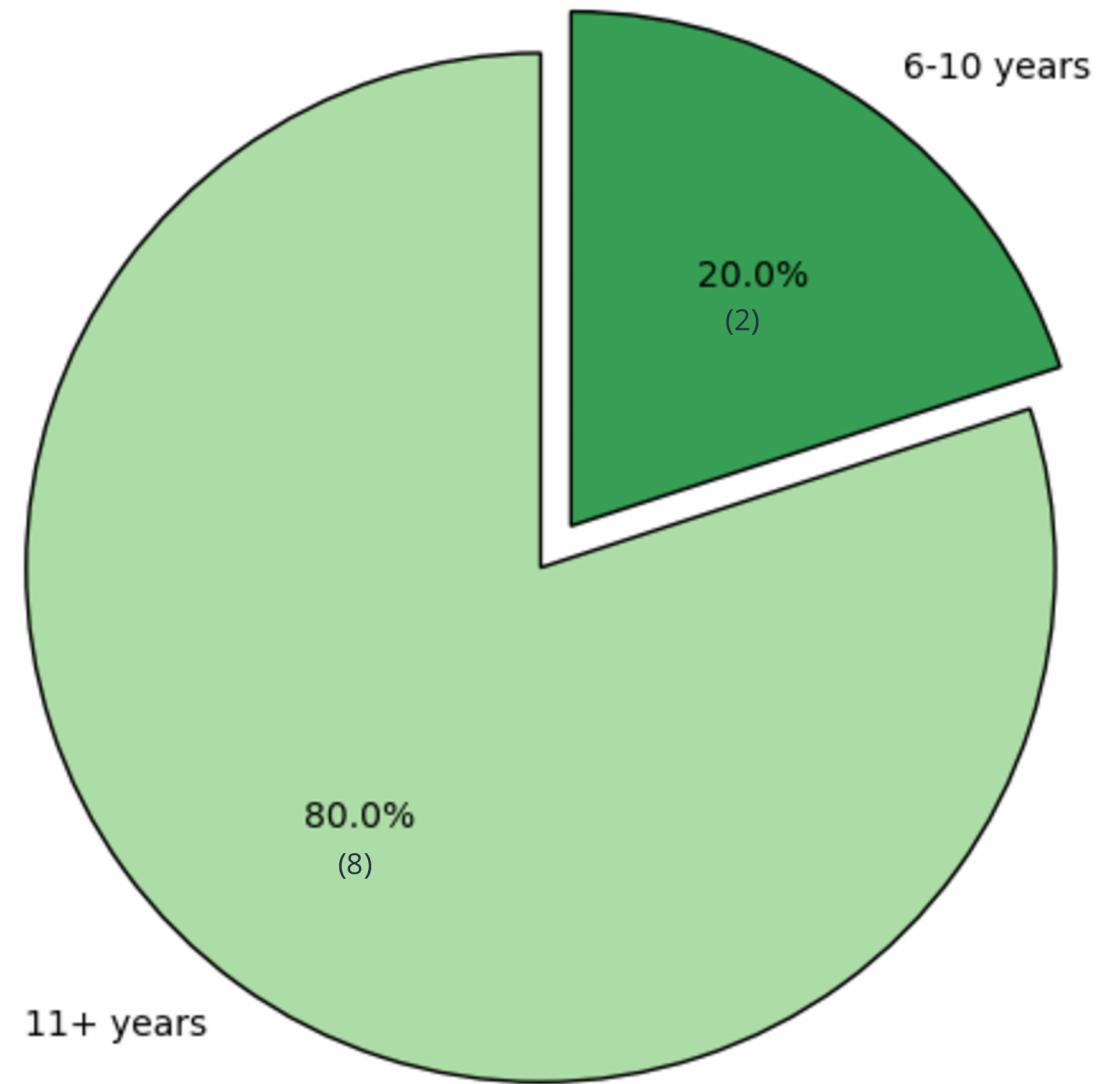
Distribution of Roles in Academia(N=12)



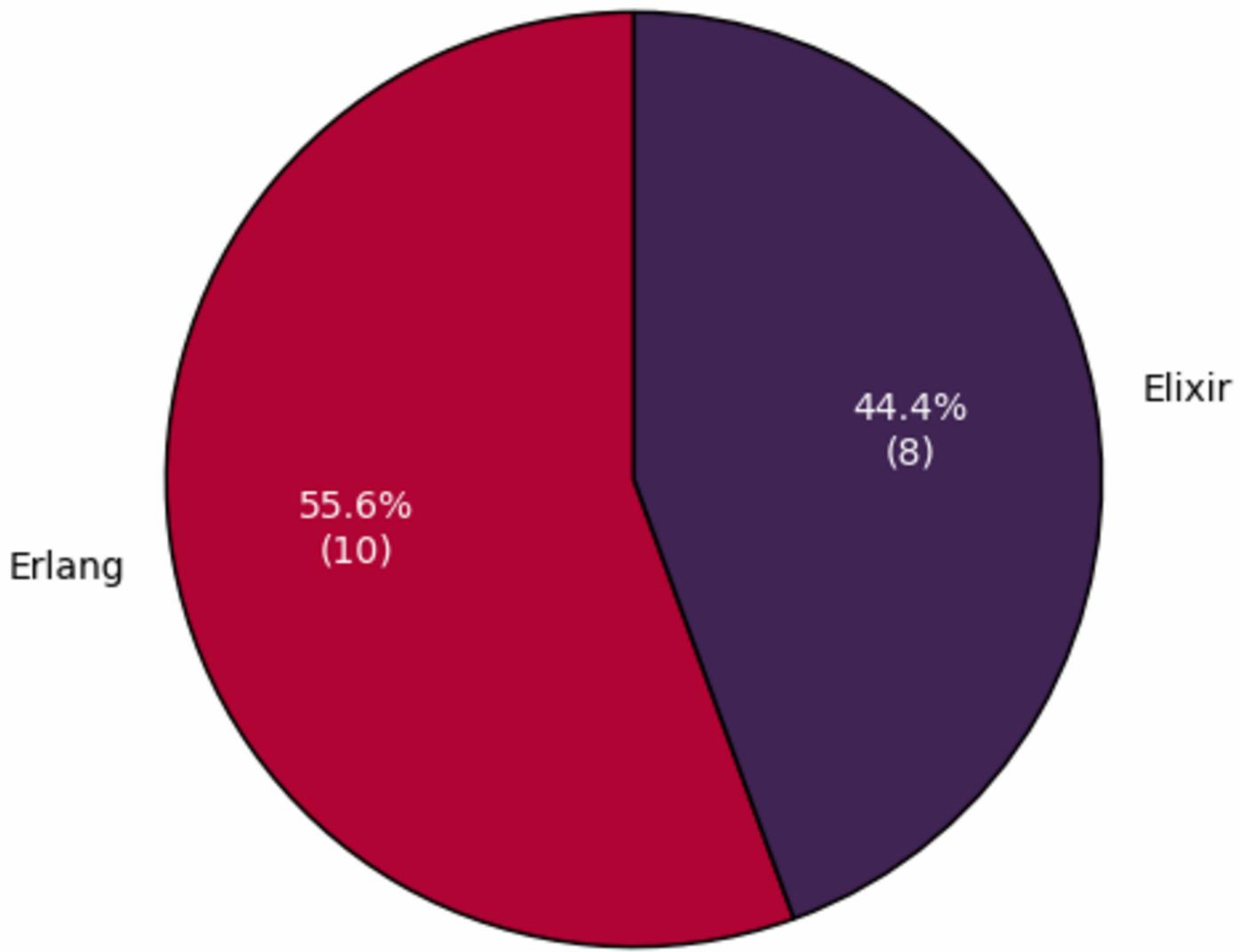
ACADEMIA



Distribution of Academy Years of experience (N=10)

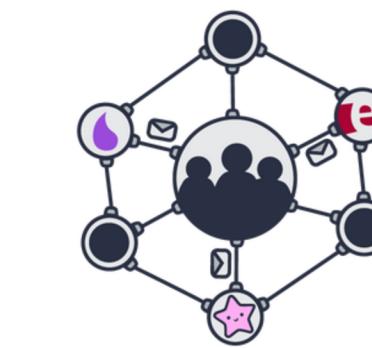


BEAM Languages Used in Research - N = 10

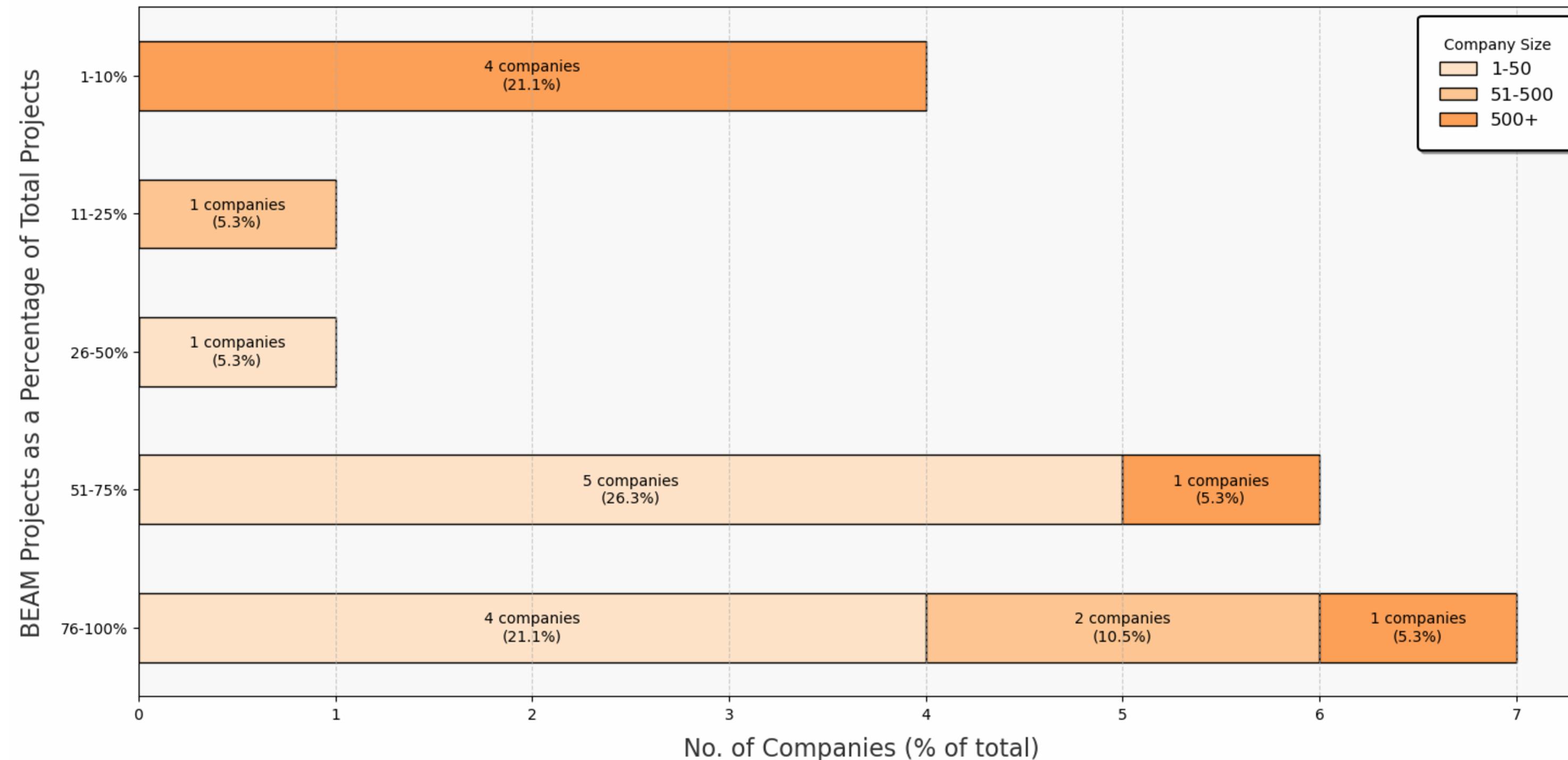


Adoption Stages

COMPANIES - ADOPTION STAGE

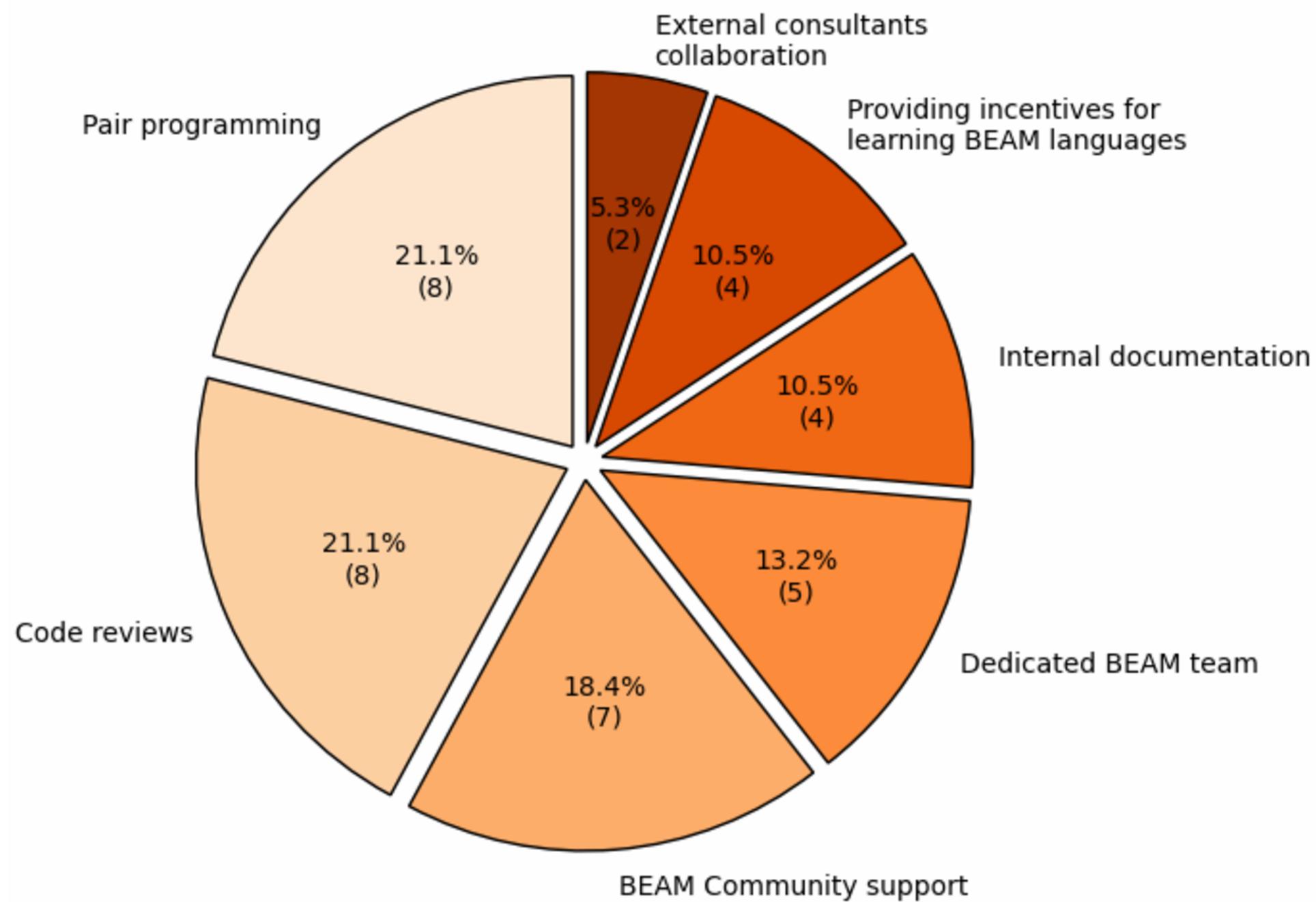
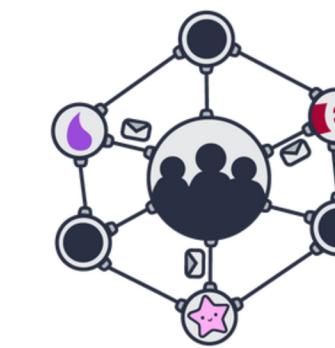


Distribution of BEAM Projects as a Percentage per Company Size, N=19



Adoption Strategies

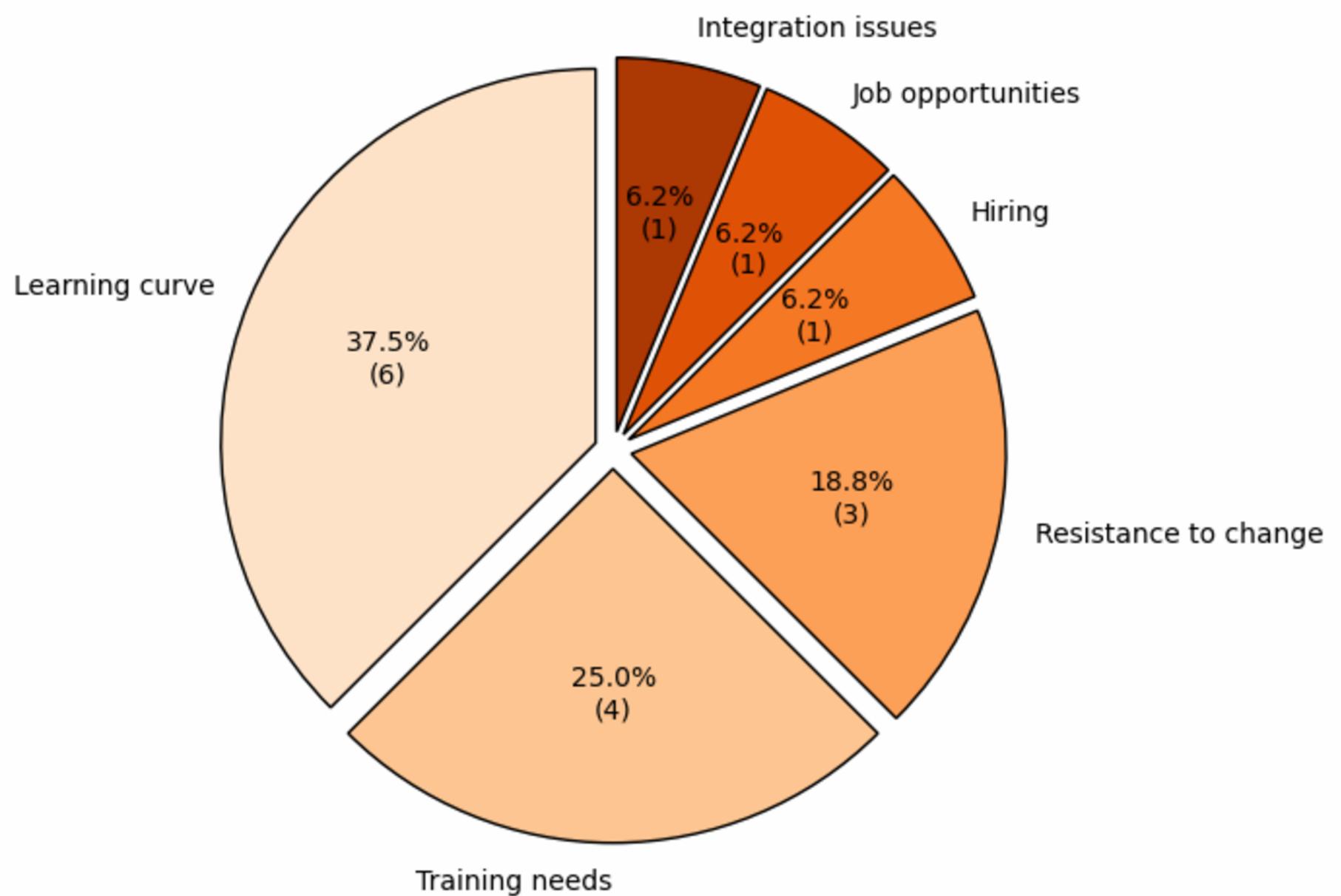
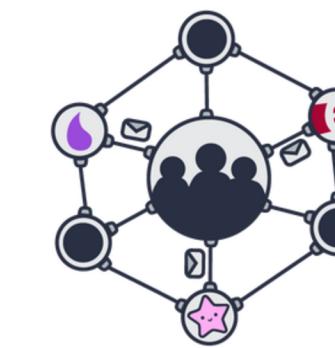
ADOPTION STRATEGIES



*Time and incentive to read the Elixir docs and the code. Experiments are amazingly useful, Elixir code is really easy to learn by doing, and to pick up patterns by simply pairing a bit and trying different things. The faster your "code, commit, deploy" loop is, the better your devs will learn. (**Company owner**)*

Adoption Challenges

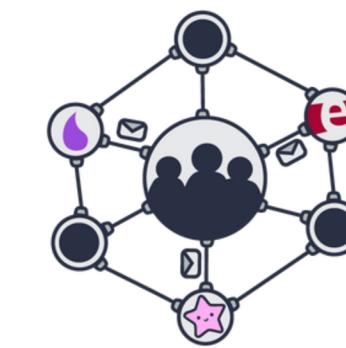
COMPANIES' CHALLENGES



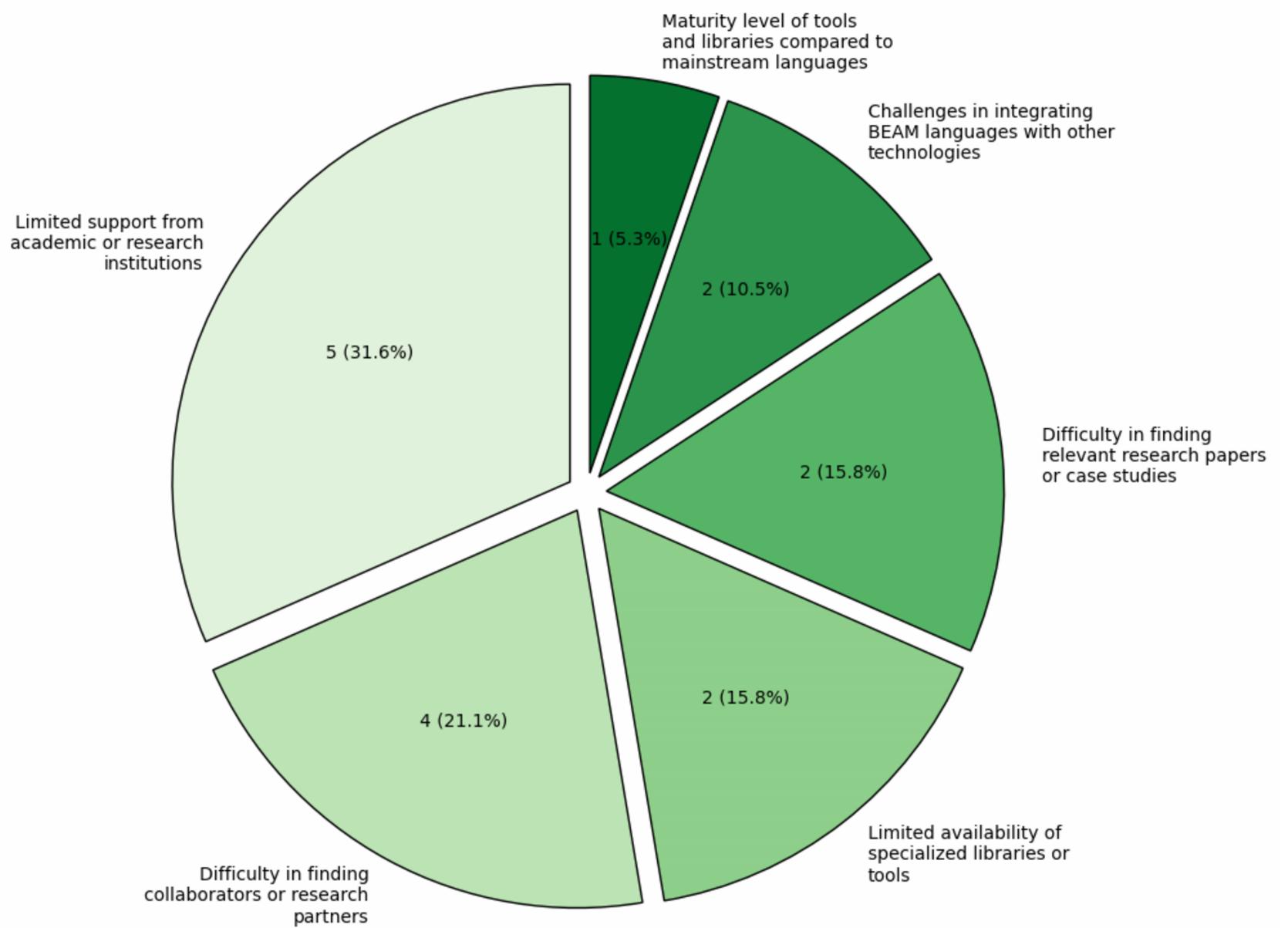
Difficulties in finding trained developers, doubts on the productivity compare to other languages (less libraries, for instance), fear to functional and concurrent paradigms.
(Academic's perspective of industry perception)

Customer doesn't "get it", observed risk of hiring difficulty, observed risk of dependency on us. Blub paradox again. Erlang's fault tolerance sounds too good to be believed, customers think its marketing bullshit when we tell them about fault tolerant software. (Company owner)
Blub Paradox - PaulGraham: <http://paulgraham.com/avg.html>

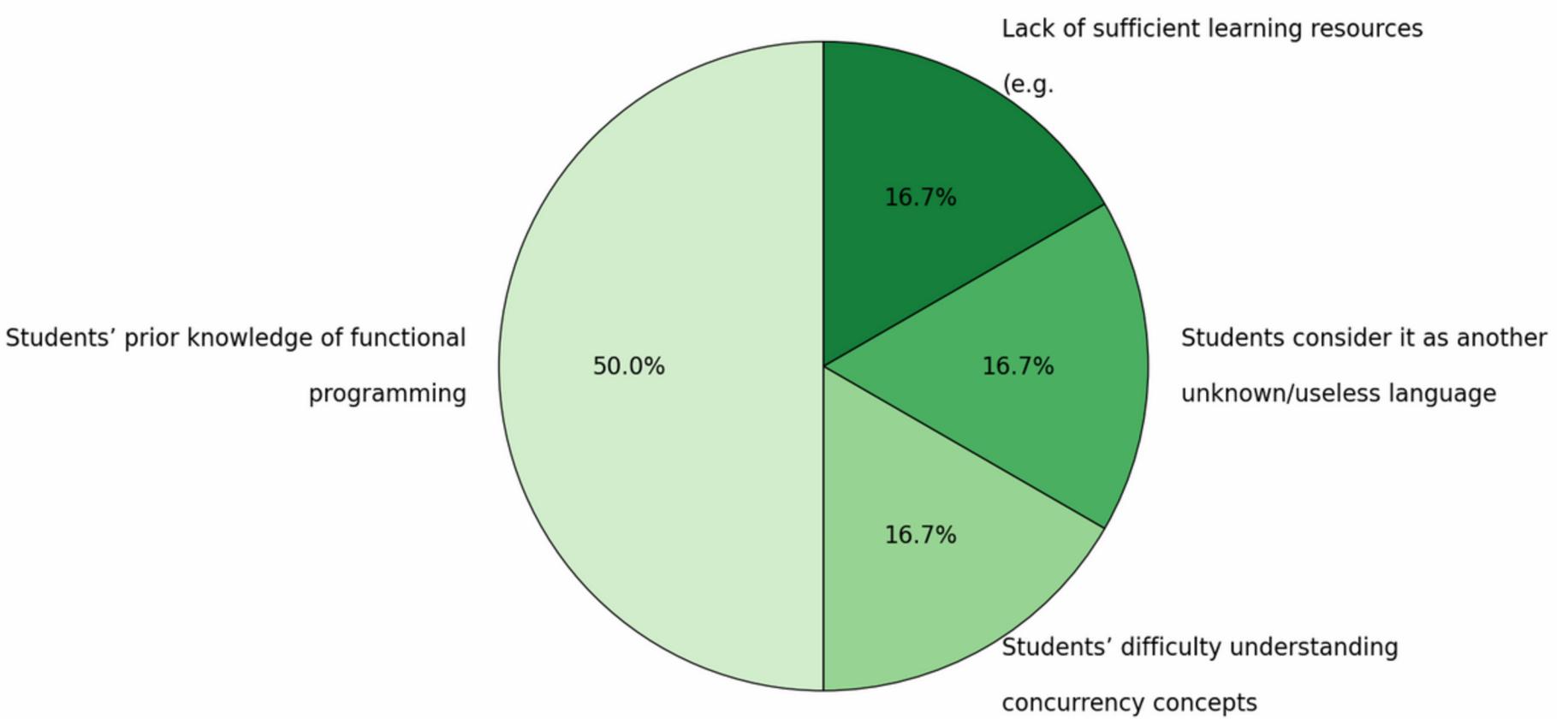
ACADEMIA'S CHALLENGES



TEACHING

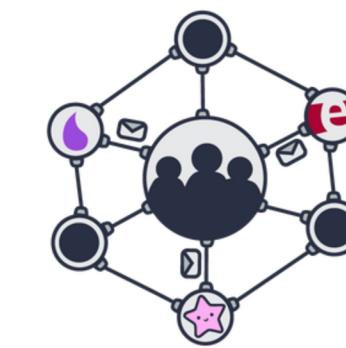


RESEARCH

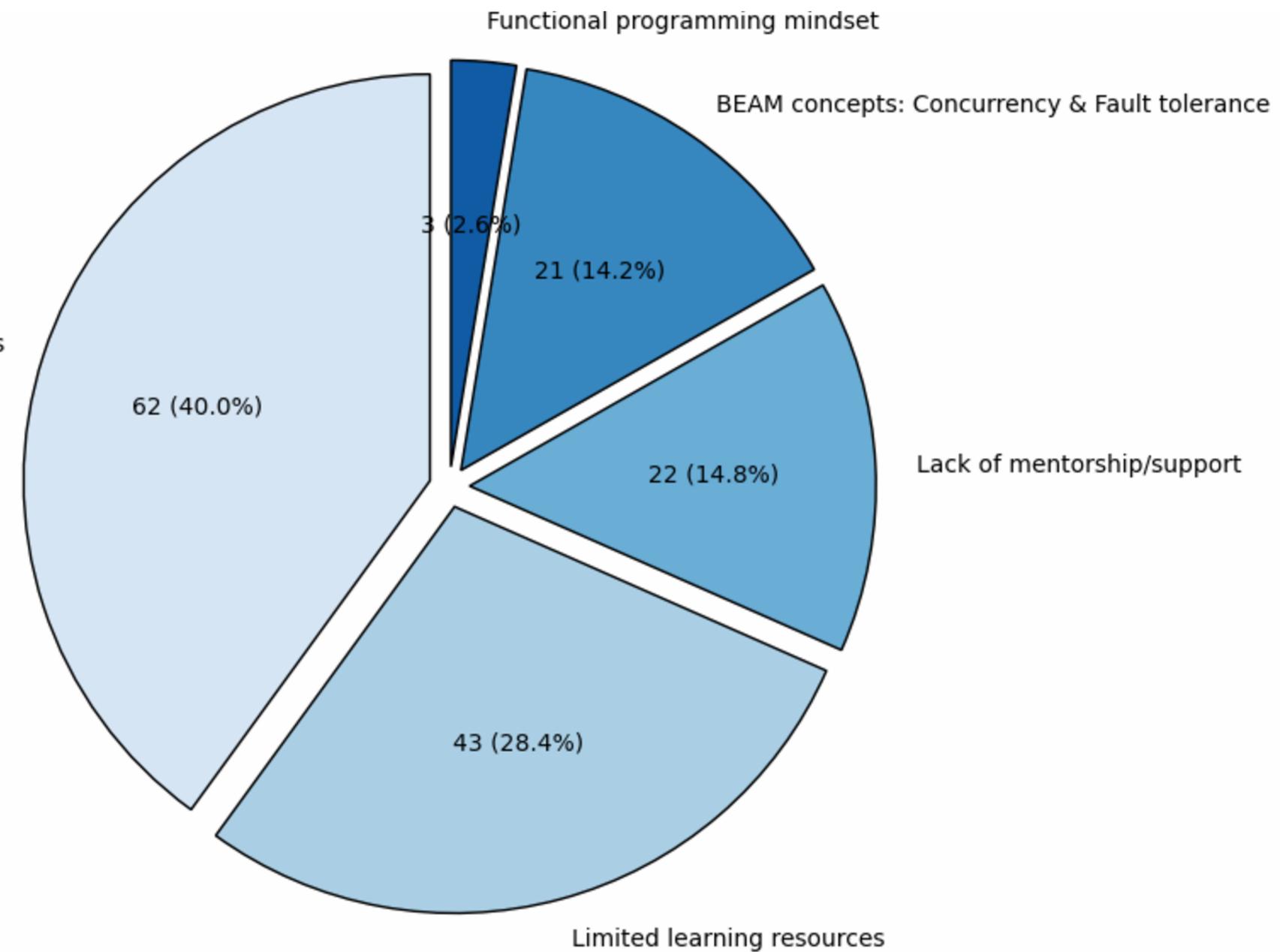


Unawareness of languages relevance (and thus additional resistance to engagement with "yet another" + "unknown = useless" language. (Academic on challenges with students)

DEVELOPERS' CHALLENGES



Challenges you face while learning BEAM languages



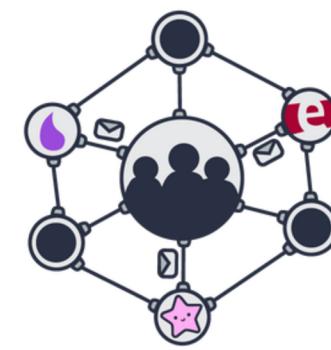
The fear of failing at learning is greater than the challenge itself. (Developer)

I would love an Gleam book, I need more resources to learn it. (Developer)

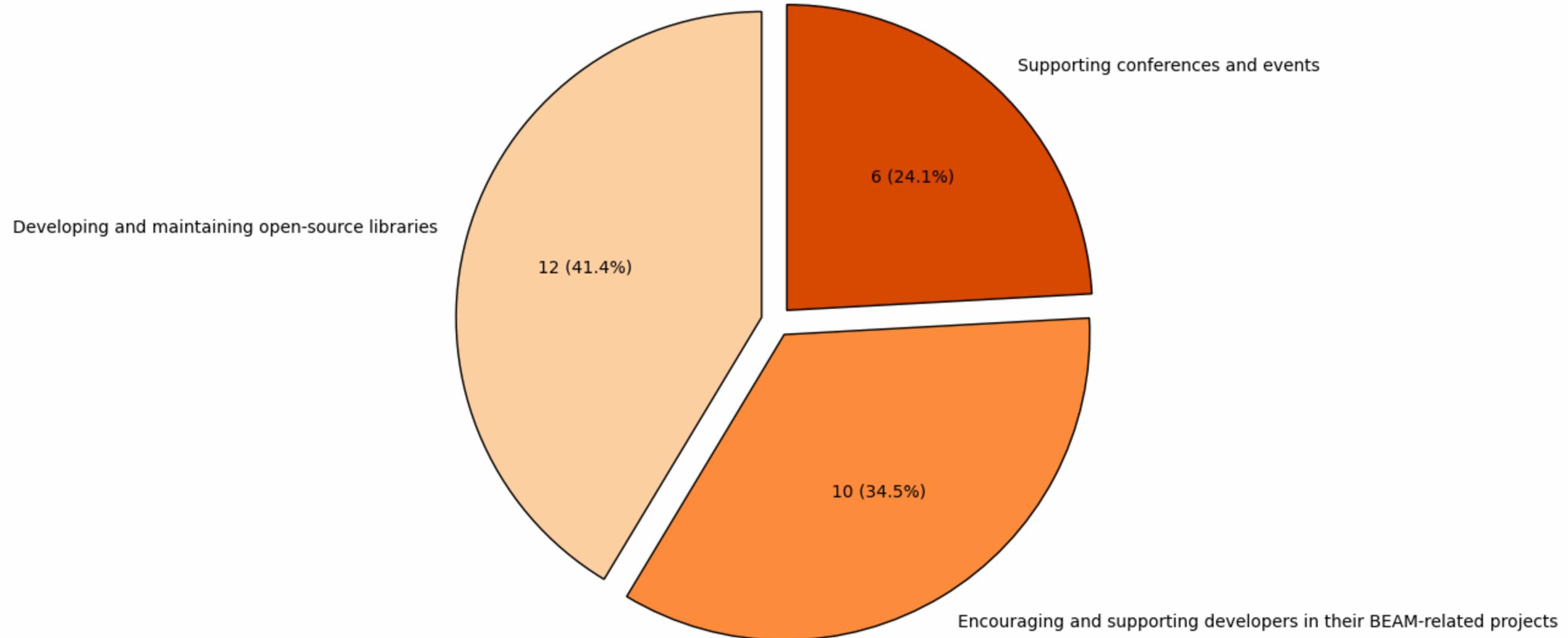
Definitely more materials how to do and configure things. For example check getting started guide for Ruby on Rails. A lot of time I know how to do things in Beam but I'm not confident that my approach is correct. (Developer)

More industry use cases being shared, more industry verticals collaborating (I know devs in healthcare seems to be working on some shared stuff). (Developer)

BEAM COLLABORATION CHALLENGES



In which ways does your company collaborate with the BEAM community?



Companies

Has your company collaborated with other companies on BEAM-related projects? 36,8%

Academia

Is your company collaborating with any academic institutions on BEAM-related projects? 17%

Would your company be interested in collaborating with academic if given the opportunity? 55%

Master thesis topics, mentorship and being one of the readers.

Education WG in EEF. Guest lectures telling students how we use Erlang in industry successfully

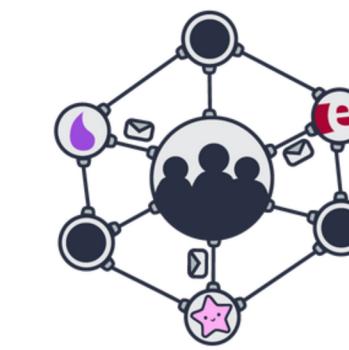
(Company)

Customer doesn't "get it", observed risk of hiring difficulty, observed risk of dependency on us. Erlang's fault tolerance sounds too good to be believed, customers think its just marketing when we tell them about fault tolerant software'

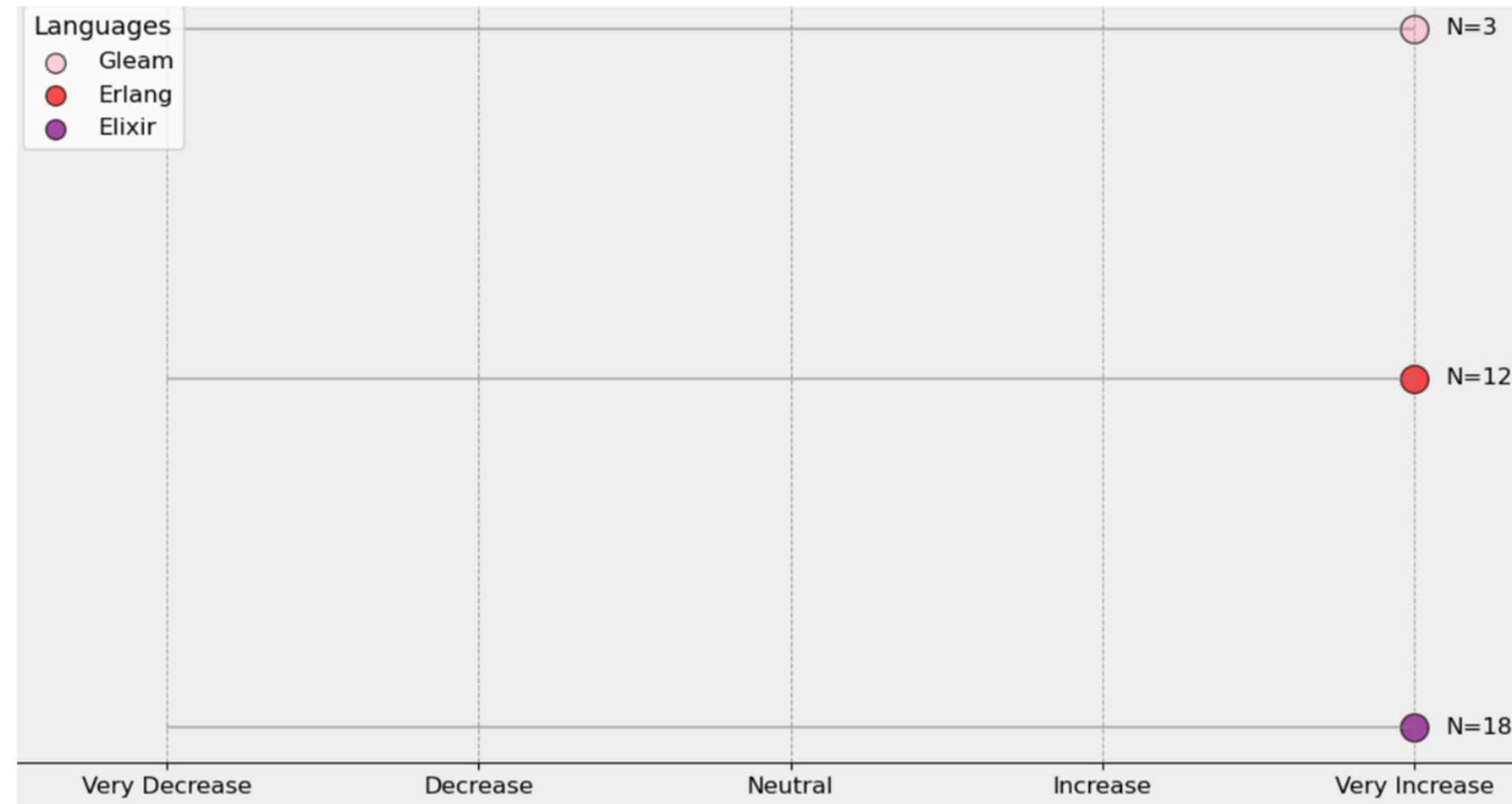
(Company)

Adoption Impact

IMPACT ON PRODUCTIVITY - DEVELOPERS



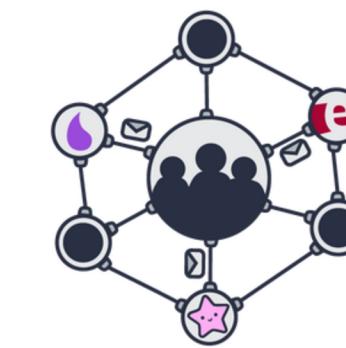
To what extent has adopting BEAM languages impacted your job **productivity**?



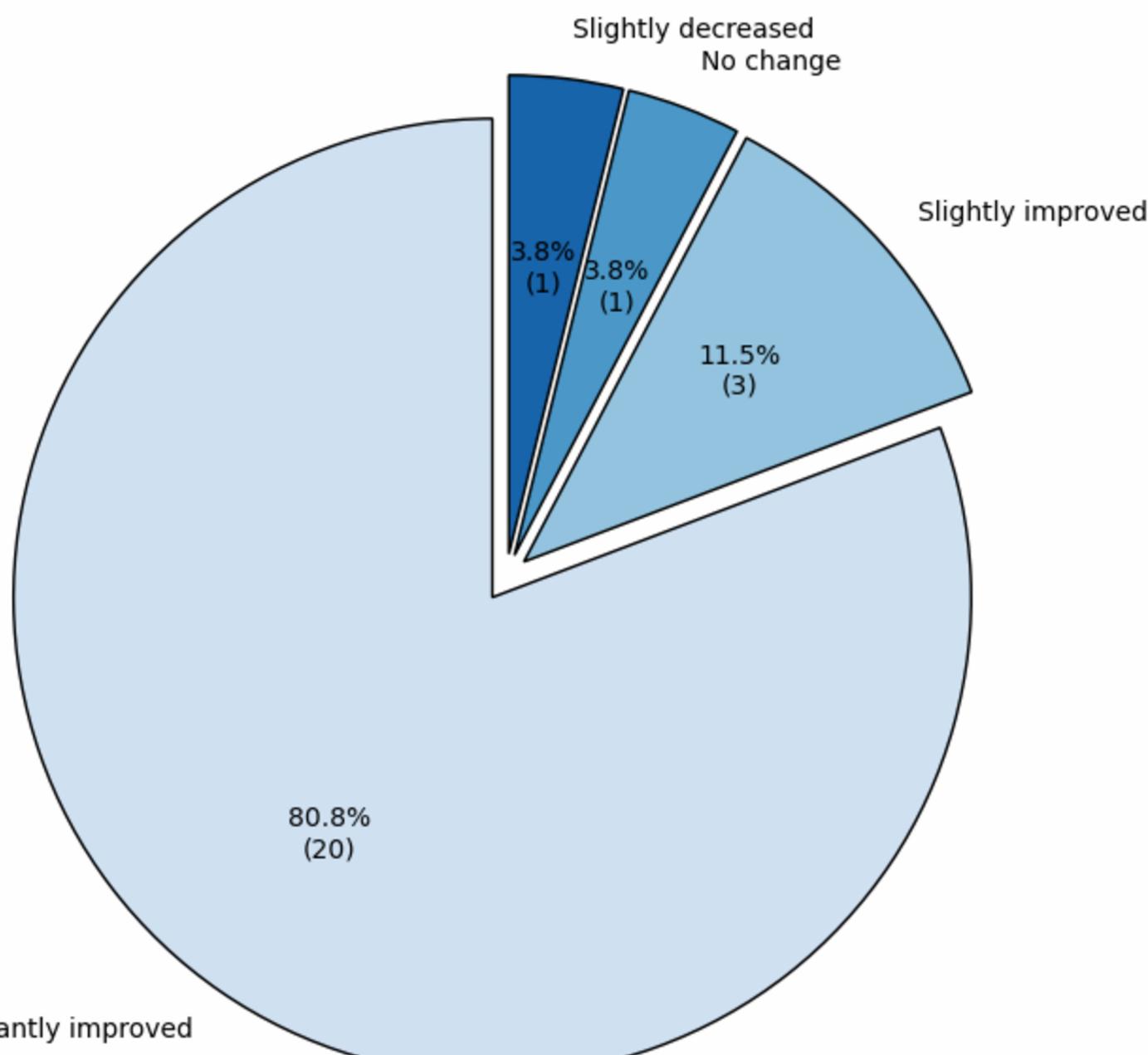
*Amazing debugging, magnificent model, beautiful language, very good framework, and so many built-in features in the otp.
(Developer)*

Being able to recompile code while the software is still running, introspection and the ability to remotely access a running instance allows me to be fast at iterating code and finding bugs. (Developer)

IMPACT ON SATISFACTION - DEVELOPERS



To what extent has adopting BEAM languages impacted your job **satisfaction**?



Erlang fits my mental model for program construction so coding is a pleasure as it feels like an extension of what I'm thinking about (vs fighting to express myself in other languages). When I'm not using Erlang, I miss it; I do not miss other languages when programming in Erlang. (Developer)

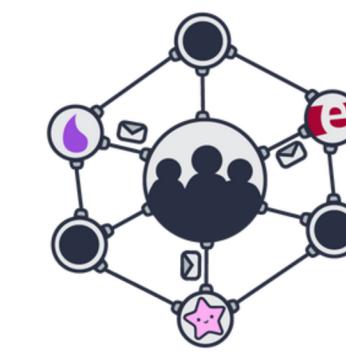
I would love to work with Elixir on a daily basis but curave a job doing Python. (Developer)

I've never felt as productive in a language as I have in Gleam, the tooling and ease of refactoring are on another level (Developer)

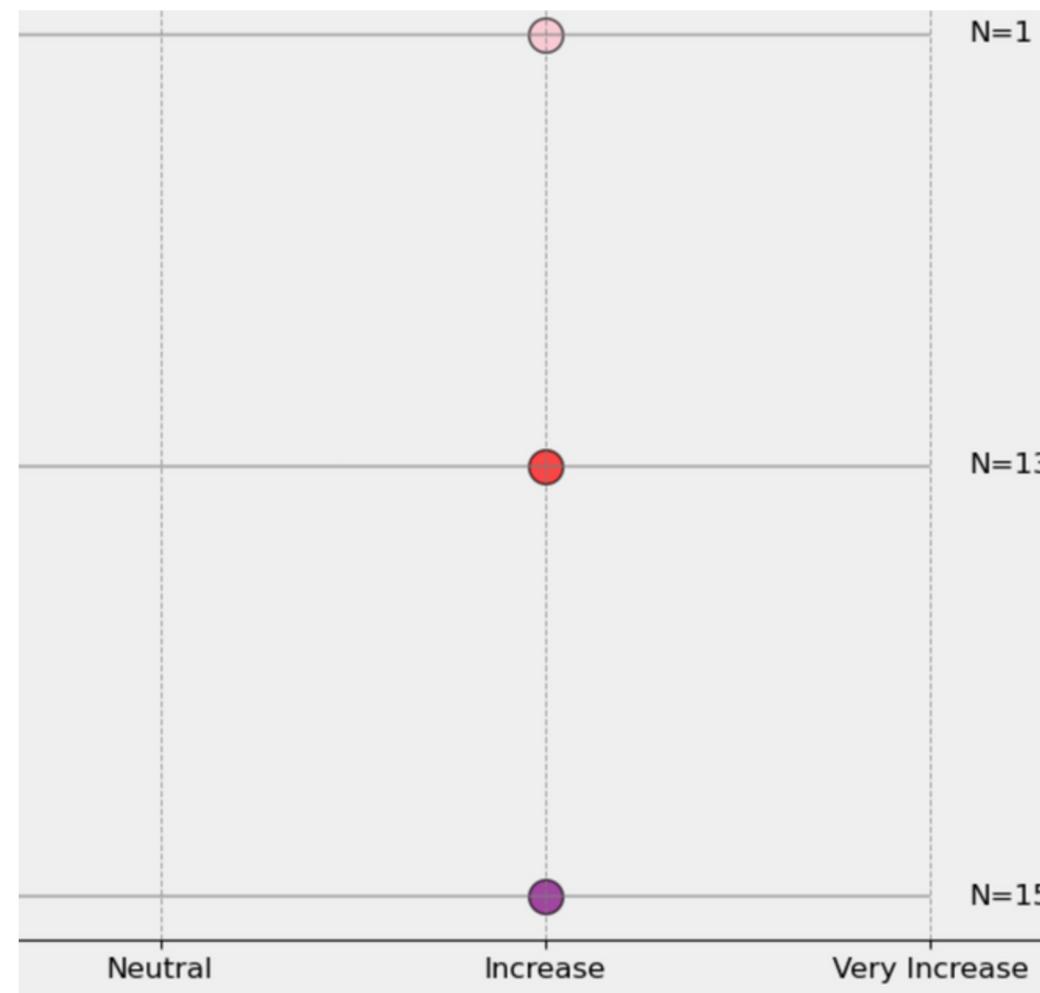
It's less fighting against the language to get things done. (Developer)

It's often times frustrating that some people don't know the whole concept of distributed systems. That makes for ugly code that breaks a lot, and is a pain to maintain. (Developer)

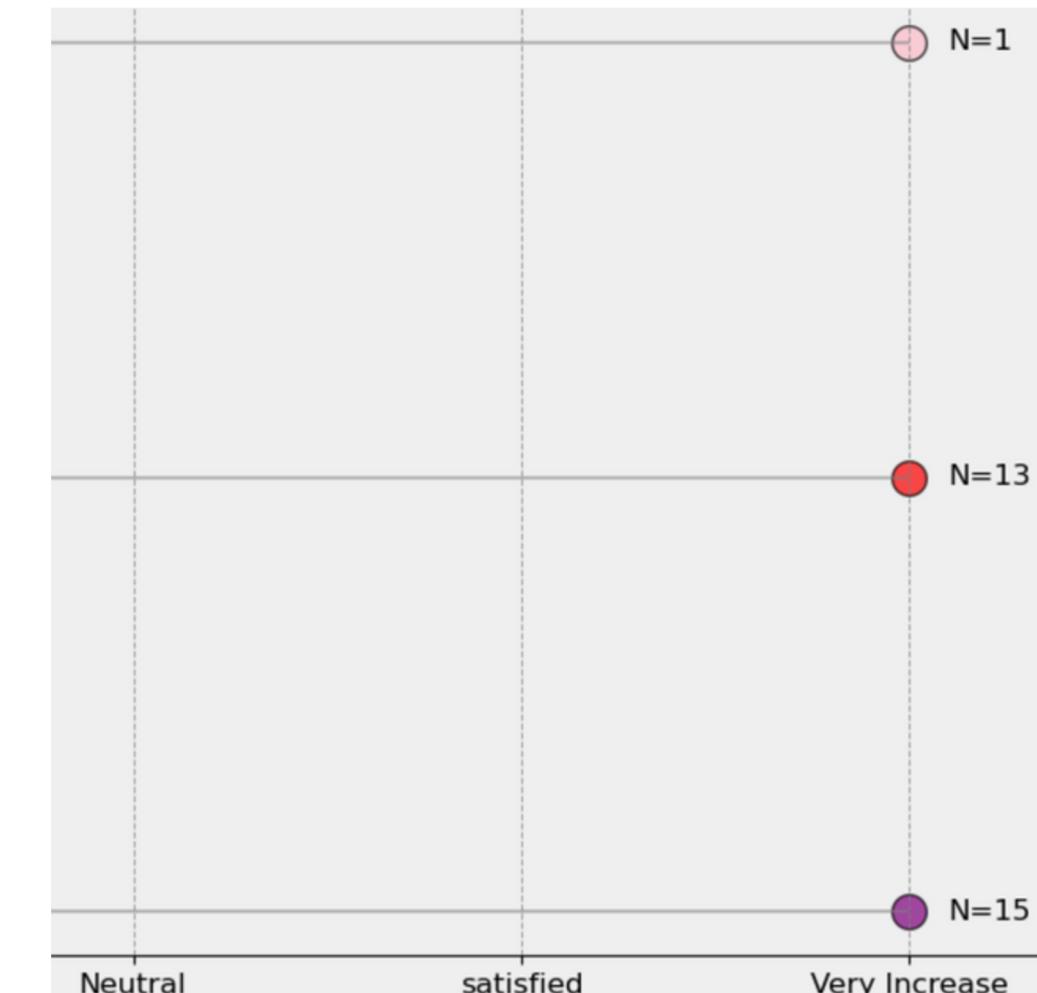
IMPACT ON INNOVATION - COMPANIES



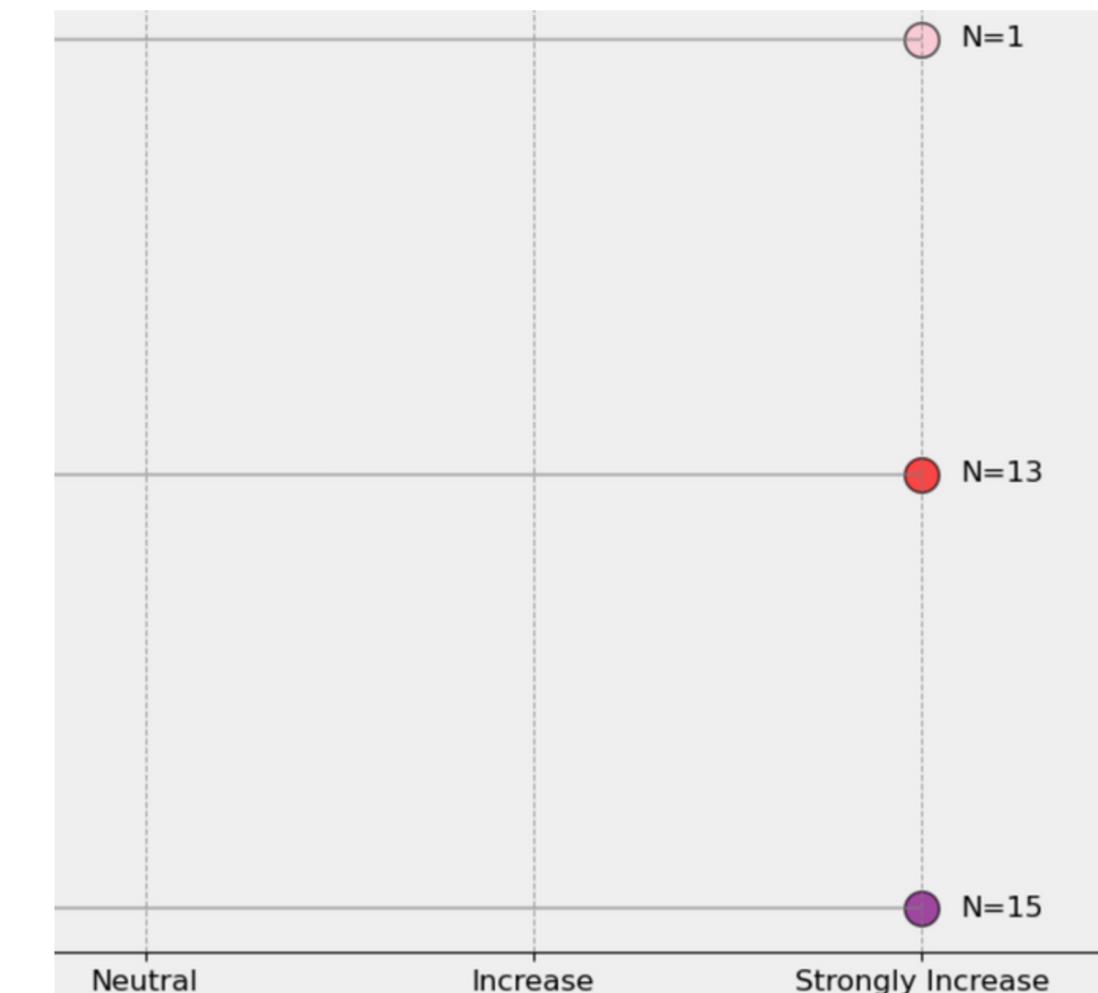
How has adopting BEAM languages influenced your company's approach to **innovation**?



What are your company's current feelings and **satisfaction** regarding the decision to use BEAM languages?

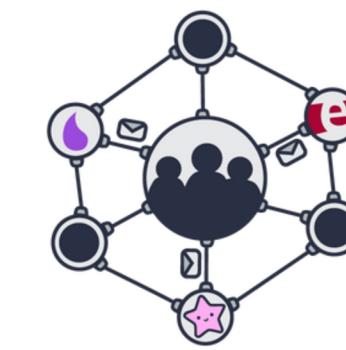


How has adopting BEAM languages affected your company's **productivity**?

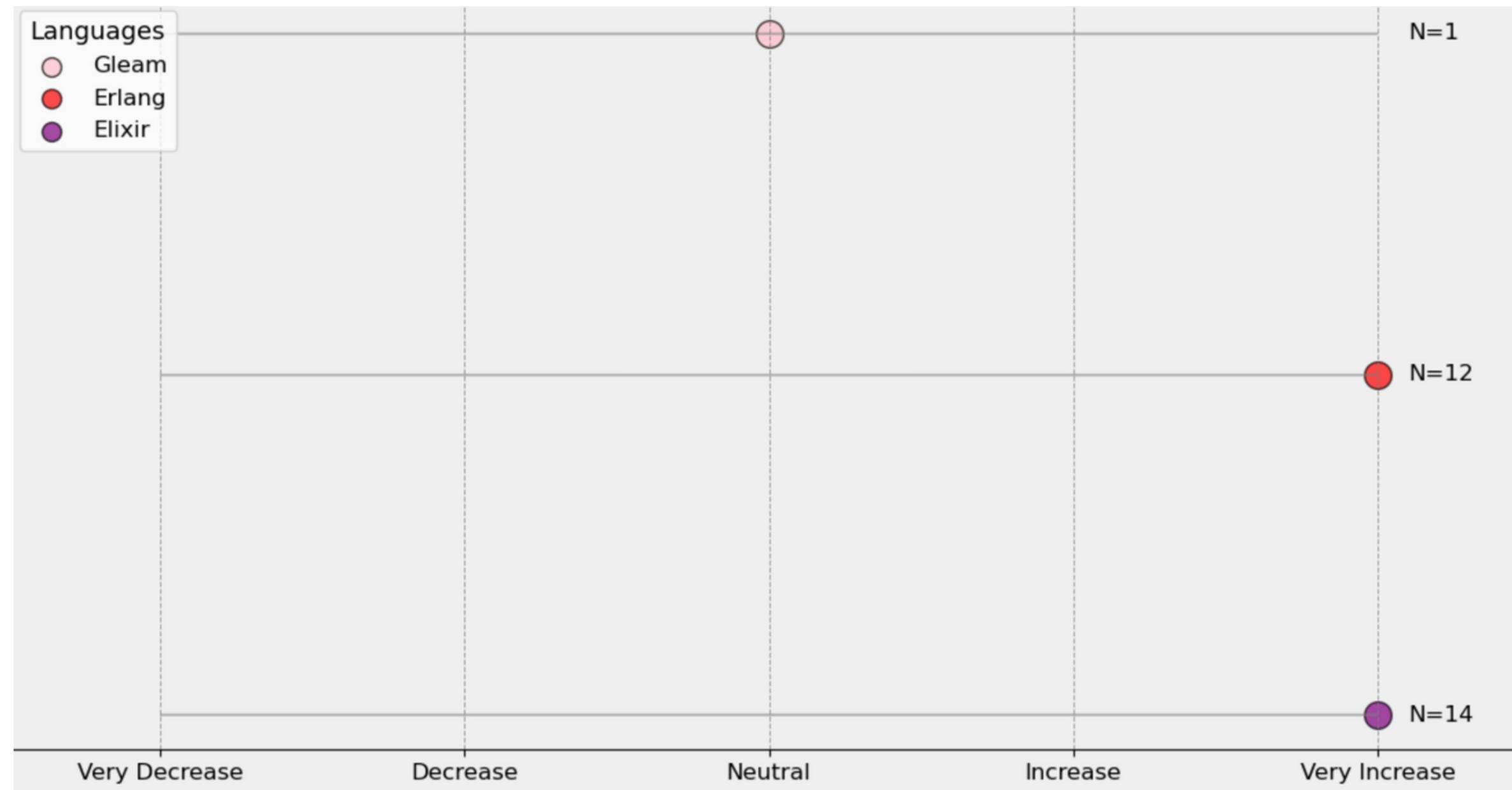


We can try complex and difficult things in a few weeks, and if it works, we know we can immediately "go wide" with it, as even "prototype" level code is reliable, scalable and easy to modify and refactor. It basically makes innovation "free".
(Company Owner on Innovation)

IMPACT ON DEV RETENTION - COMPANIES

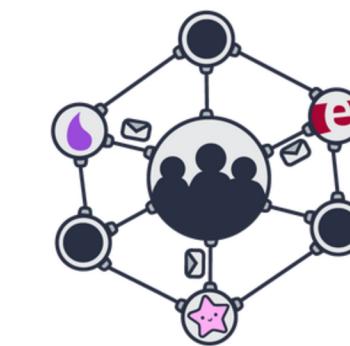


How has adopting BEAM languages affected your company's ability to retain skilled developers?

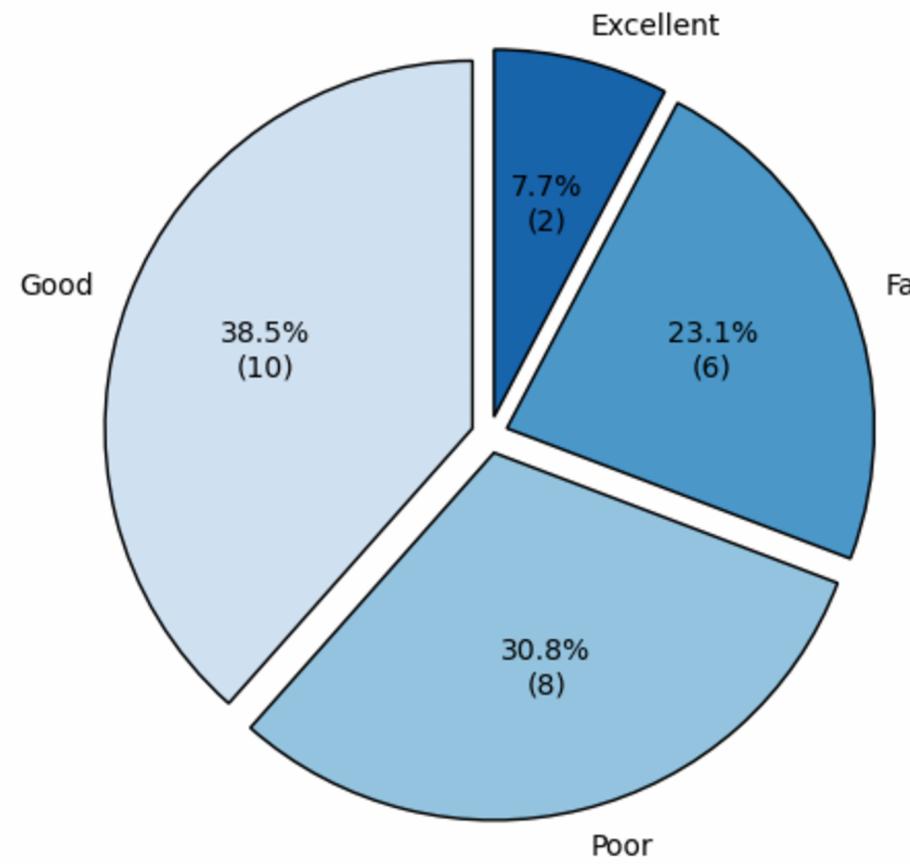


Mental Health

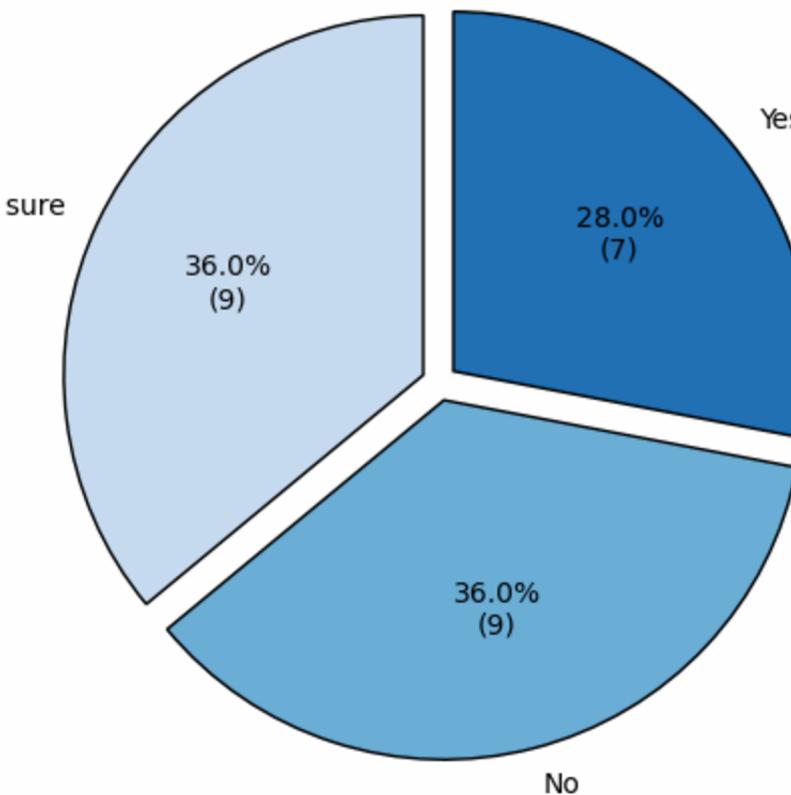
MENTAL HEALTH & WELL-BEING - COMPARING DEVELOPER AND COMPANY PERSPECTIVES



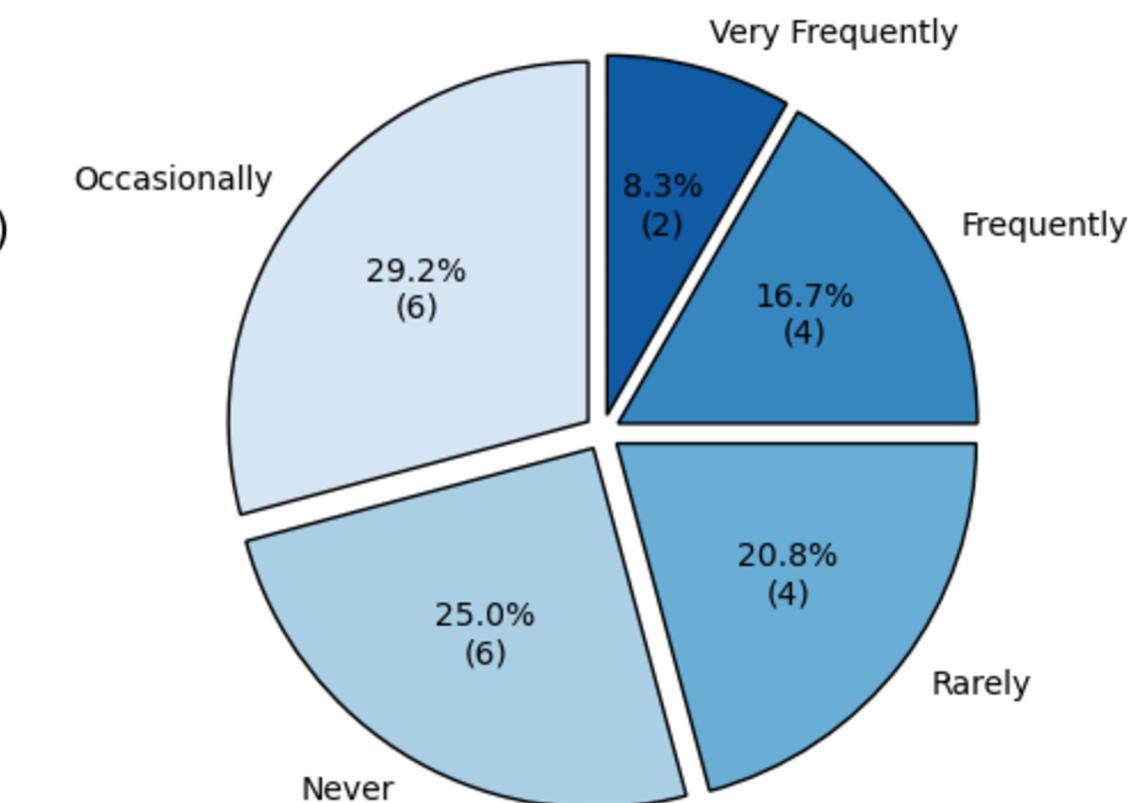
How would you rate your overall mental well-being? (n=26)



Do you have any mental health conditions? (n=25)

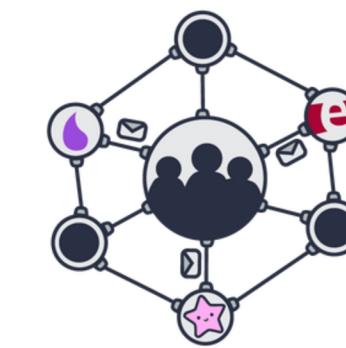


How often do you experience stress
or burnout due to your work as a BEAM developer?
(n=24)

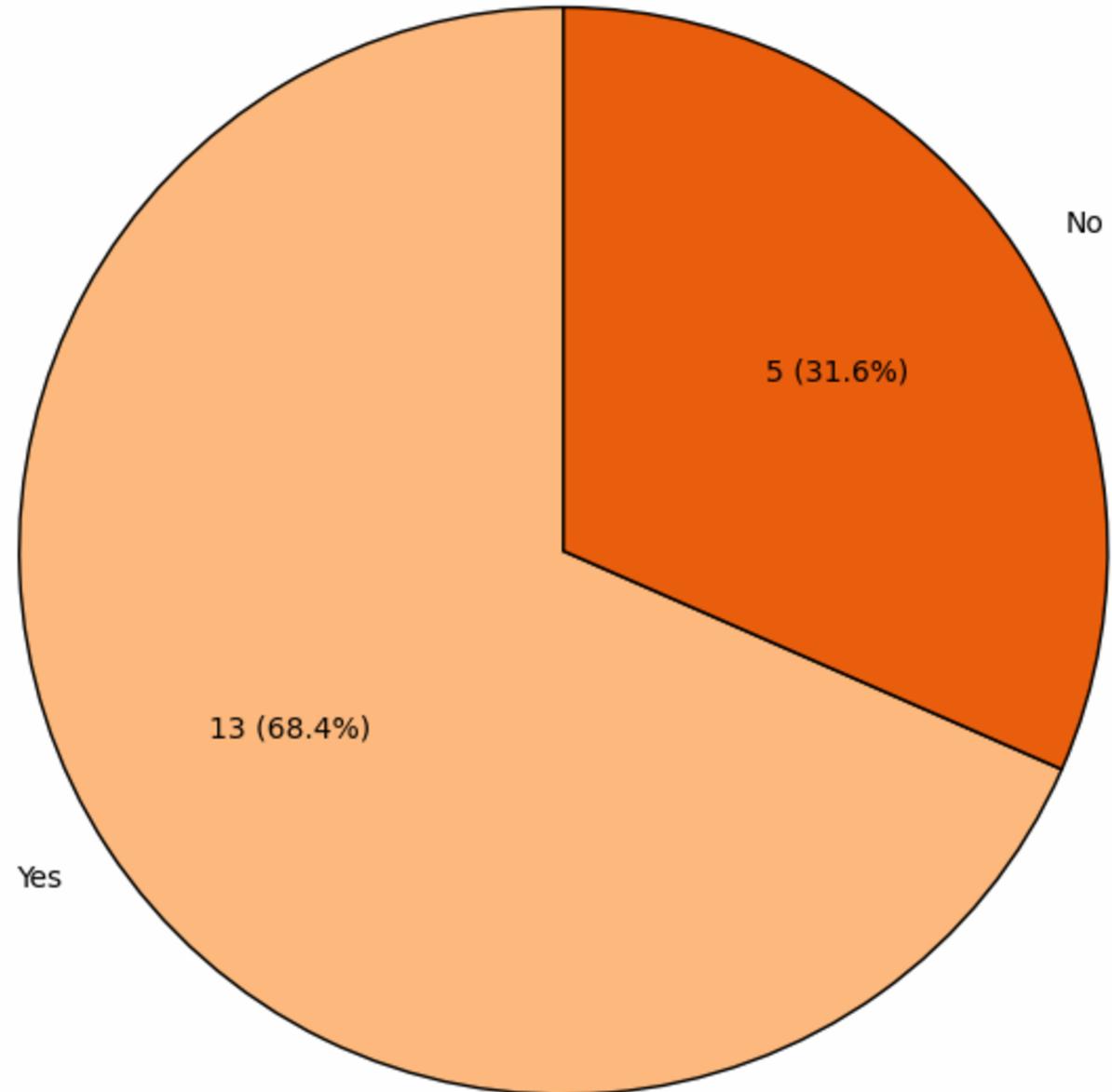


Source for reference: "Mental Health in the Workplace: a global picture", Global Business Collaboration for Better Workplace Mental Health (GBC)
<https://betterworkplacemh.com/>

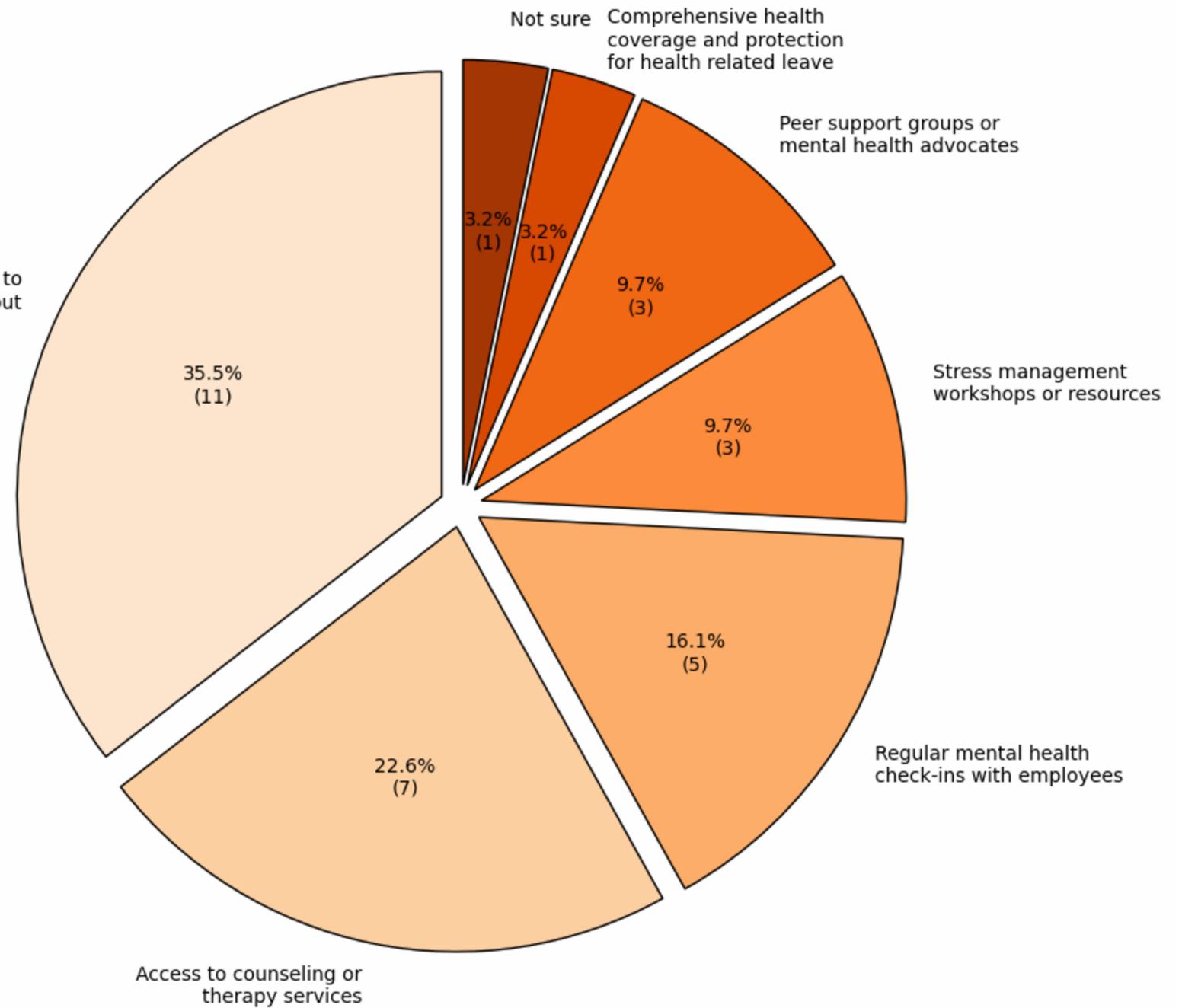
MENTAL HEALTH & WELL-BEING - COMPARING DEVELOPER AND COMPANY PERSPECTIVES



Does your company have practices to prevent mental health issues?

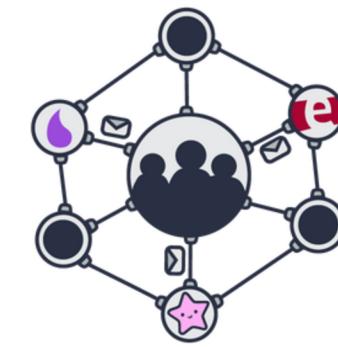


How?



Opportunities: Exploring Solutions and Improvements

OPPORTUNITIES & IMPROVEMENTS - WHAT'S NEXT?



More exposure to solutions made using BEAM and BEAM languages to other groups outside of the BEAM community itself. (Developer)

What we are missing is better documentation of the runtime. The documentation is very fuzzy, but for research and teaching, we need to know the details. However, the best resource for understanding the runtime seems to be the implementation - and I can't find the time to sit down and try to reverse-engineer the semantics from the runtime code. (Academic)

I want to have a closer look at Livebook because it seems a great tool for using in teaching. A resource that my students are missing is stackoverflow (there are only few entries related to BEAM languages). While it might be useful to have more support/solutions there regarding installation and setup, I like it actually that there is little to be found on solutions to exercises.... (Academic)

I think it would be great to have some larger companies adopting / supporting / vocalizing their usage of BEAM languages. (Developer)

Better interaction across different languages. (Developer)

The community is not growing as fast as I'd like. We have a good number of developers interested in the ecosystem, but not as many companies or jobs around, which makes things hard. (Developer)

There is a stagnation in the BEAM related positions on the Marketplace. This is the negative thing I'm seeing and this should improve in some way. Business are not adopting BEAM tech... (Developer)

Any questions ?

Thanks for your time!