Coqdev

Recent news on making the Coq ecosystem accessible, robust to evolution, and organizing its community

Théo Zimmermann

Inria

June 1st, 2022

Personal introduction

- Member of the core development team of Coq:
  - First contribution to Coq in 2015 (8.5).
  - Participated in the release management since 2017 (8.7).
  - Maintainer of the documentation since 2018 (8.8).
  - Founder of the Coq-community organization.
  - Main developer of coqbot, a bot supporting Coq development.
- Defended a PhD at Université de Paris in 2019:

  "Challenges in the collaborative evolution of a proof language and its ecosystem"
- Inria employee (funded thanks to Nomadic Labs) since 2020.

Coqdev project: overview

- Ongoing project (2020-2022).
- Funds my 3-year fixed-term Inria position.

Objectives:

- Accessibility of the Coq ecosystem.
- Community organization and development.
- Making Coq evolution robust.

"Scaling Coq by tapping into the incredible ressources of its user community."

Methods:

- Serving as release manager, community manager, maintainer...
- Open source tools to help Coq developers and Coq projects.
- Empirical studies and literature search.

How is Coq developed today?

- Core development team: currently 10 Inria employees (in Paris, Nantes and Sophia-Antipolis) + 1 US member.
- Development in the open:
  - **Written communication** on GitHub + other public platforms.
  - **Weekly Coq Calls** open to all / notes posted on the wiki.
- All code changes must be reviewed (pull requests).
- Maintainer teams for each component ($>$ 30 maintainers).

Figure 1: Coq's current 11 core developers

1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels
2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations
3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels
2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations
3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels
2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations
3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

What is the Coq Platform?

- Distribution of Coq with curated selection of:
  - libraries
  - plugins
  - tools
- Release 2022.04.0 contains up to **50 packages** for 4 Coq versions.
- Objectives: a distribution of Coq for developing and teaching with Coq that is:
  - Operating-system independent
  - Dependable
  - Easy to install
  - Comprehensive
- Partial solution to **dependency hell**, reproducibility, and maintainability in Coq projects.

Using the Platform

- Binary installers (Windows, macOS, Snap)
  - fast
  - simple
  - not customizable
- Interactive scripts (Windows, macOS, Unixes)
  - slower
  - lower level
  - customizable

History

- 2014: Coq packages are distributed with **opam**.
- 2017: opam on Windows is ☹, so we start shipping external packages with Coq's **Windows installer**.
- 2019: Michael Soegtrop drafts the Coq Platform's **charter**.
- First half of 2020: with Yann Régis-Gianas, I supervise Antonin Décimo, on building Platform **infrastructure** for OCaml/Coq.
- End of 2020:
  - Michael Soegtrop releases the **first version** of the Coq Platform.
  - Enrico Tassi drafts CEP #52 which makes the Coq Platform the **official mechanism** for releasing Coq to users.
- 2021: Coq Platform releases support **several Coq versions**.
- 2022: With Karl Palmskog and Enrico Tassi, we wrote a paper that I presented at the **ETAPS workshop** "Reproductibility & Replicability of Research Results" (RRRR).

1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels
2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations
3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

Context

- Pierre Castéran has created a **new book**, along with a Coq development, to teach proof techniques.
- We have created the Coq-community organization together in 2018 with collaborative **documentation writing** as one of its objectives.
- Thus, the repository containing the code of the book and the Coq development has joined Coq-community.
- The book contains **code snippets** but these were manually synchronized with the Coq development.

Solution

- In 2020, Clément Pit-Claudel released **Alectryon**, a tool to create documentation from Coq code interleaved with rich text comments.
- In 2021, I supervised the internship of Jérémy Damour. Objectives:
  - Coq contributions to Hydras & Co.,
  - building an Alectryon-based toolchain to automatically **synchronize code snippets** with the Coq code.
- The book has been **entirely converted** to this new toolchain.
- Others have since then used our pipeline, e.g., to prepare slides.
- Eventually, this should become a feature of Alectryon.
- With Pierre, Clément, Jérémy, and Karl Palmskog, we have published a paper that we will present soon at **JFLA 2022** about this project and its use of modern maintenance tools.

1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels
2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations
3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

Available channels

- Since long ago: Coq-Club mailing list
- Since 2019 (after OCaml): Discourse forum
- Since 2020 (after Lean): Zulip chat
- Since 2022 (started by Andrej Bauer): Proof Assistants SE

Integration:

- New Discourse posts and questions on all Stack Exchange websites with **tag coq** are relayed to Zulip.

Results:

- Coq-Club is still active and Discourse has not taken off.
- Zulip is a **big success**.
- Proof Assistants SE has taken off initially under the impulse of Lean users, but Coq is now the **most popular tag**.

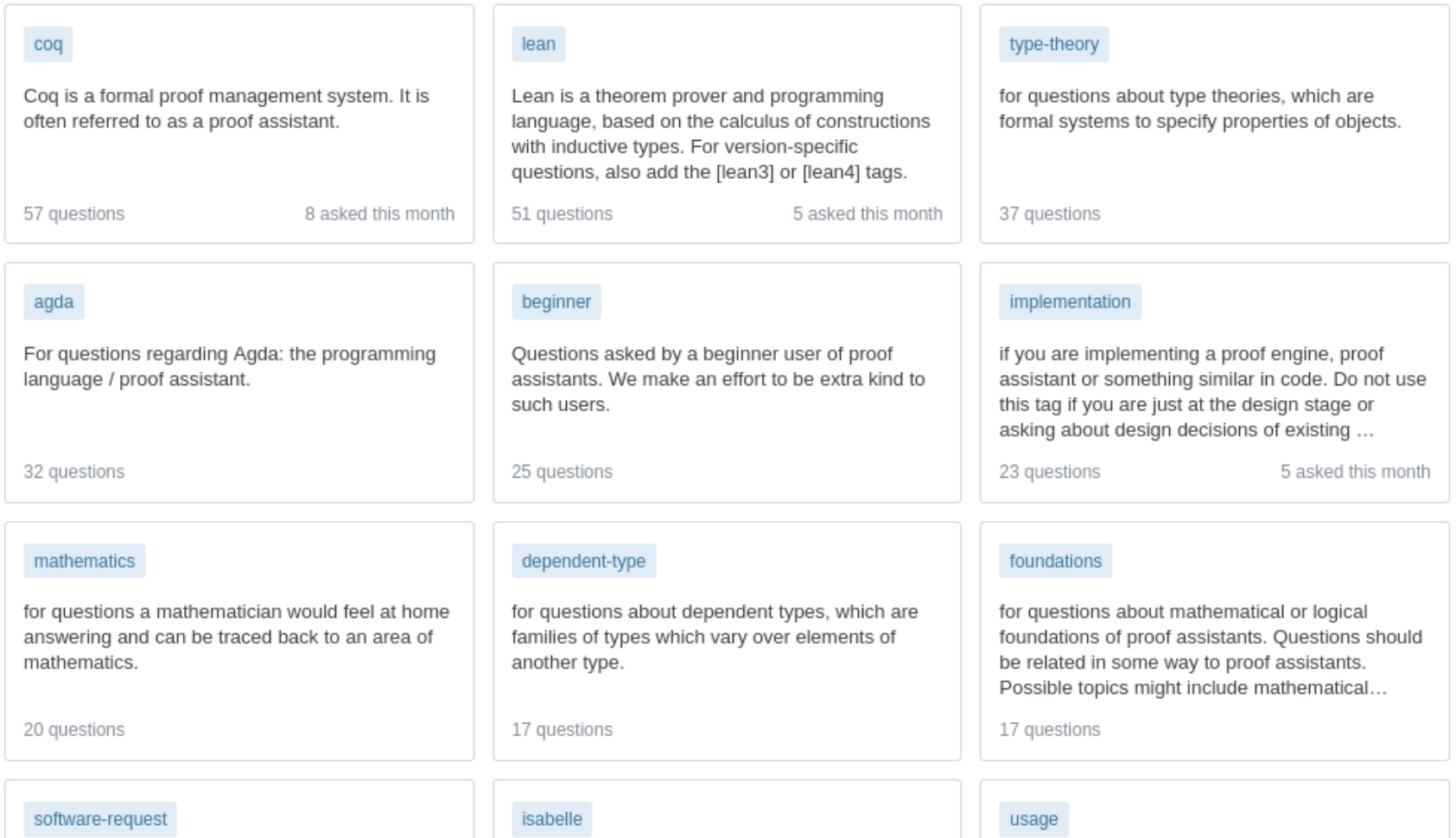


Figure 2: Popular tags on Proof Assistants Stack Exchange

1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels

2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations

3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels
2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations
3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

Objectives

The Coq Community Survey was inspired in part by the OCaml Community Survey, with the objectives:

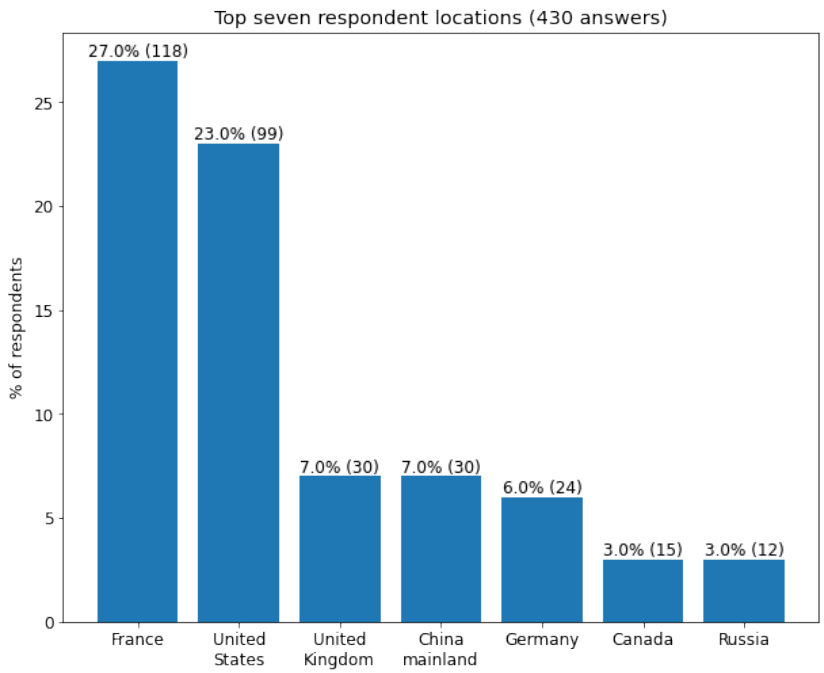
- to get an **updated picture** of the Coq community,
- to get input from the community to **inform future decisions** (about features, where to focus efforts, but also whether to rename Coq).

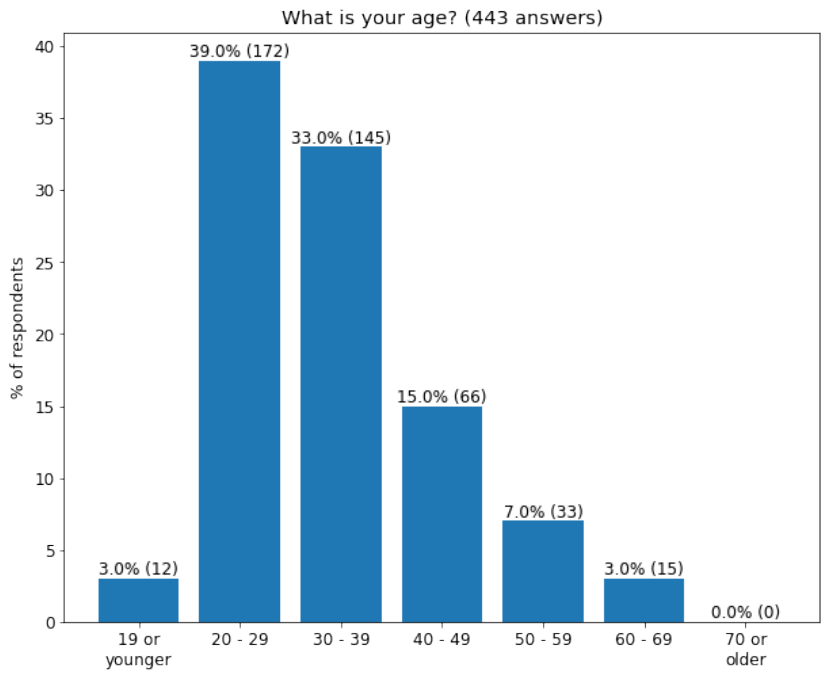
Coq Community Survey WG

- I have formed an **ad hoc WG** to prepare the survey after an open call to participation.
- The WG includes:
  - some Coq core developers,
  - members of the **Coq community**,
  - software engineering **researchers from my network**, who are more experienced with preparing and analyzing surveys.
- The WG has met almost every week to prepare the survey starting in October.
- The survey was held during the month of **February**. It was available in English and in Chinese.
- Since then, we have been analyzing the results.

Results

Results will be shared with the community in a series of **blog posts** (the first one should come soon) and eventually in a research paper.





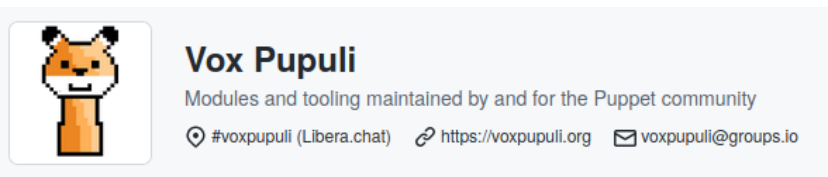
1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels
2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations
3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

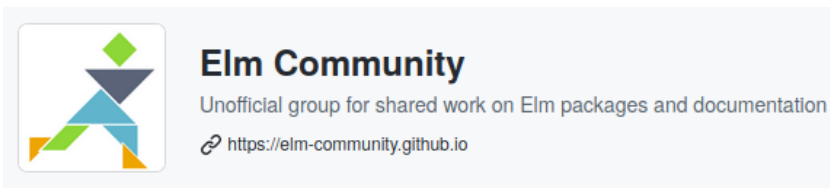
Context

- During my PhD, I have observed a model of **Community Package Maintenance Organizations** (CPMOs) in several ecosystem.
- I have applied this model to the Coq ecosystem by founding the **Coq-community** organization.
- In 2021, Jean-Rémy Falleri (SE researcher from LaBRI) and I have decided to better understand this model by performing a **qualitative study**.

What are CPMOs?

- Initiatives by **users**
- of a specific **ecosystem**
- to prevent package abandonment
- by **“adopting”** packages

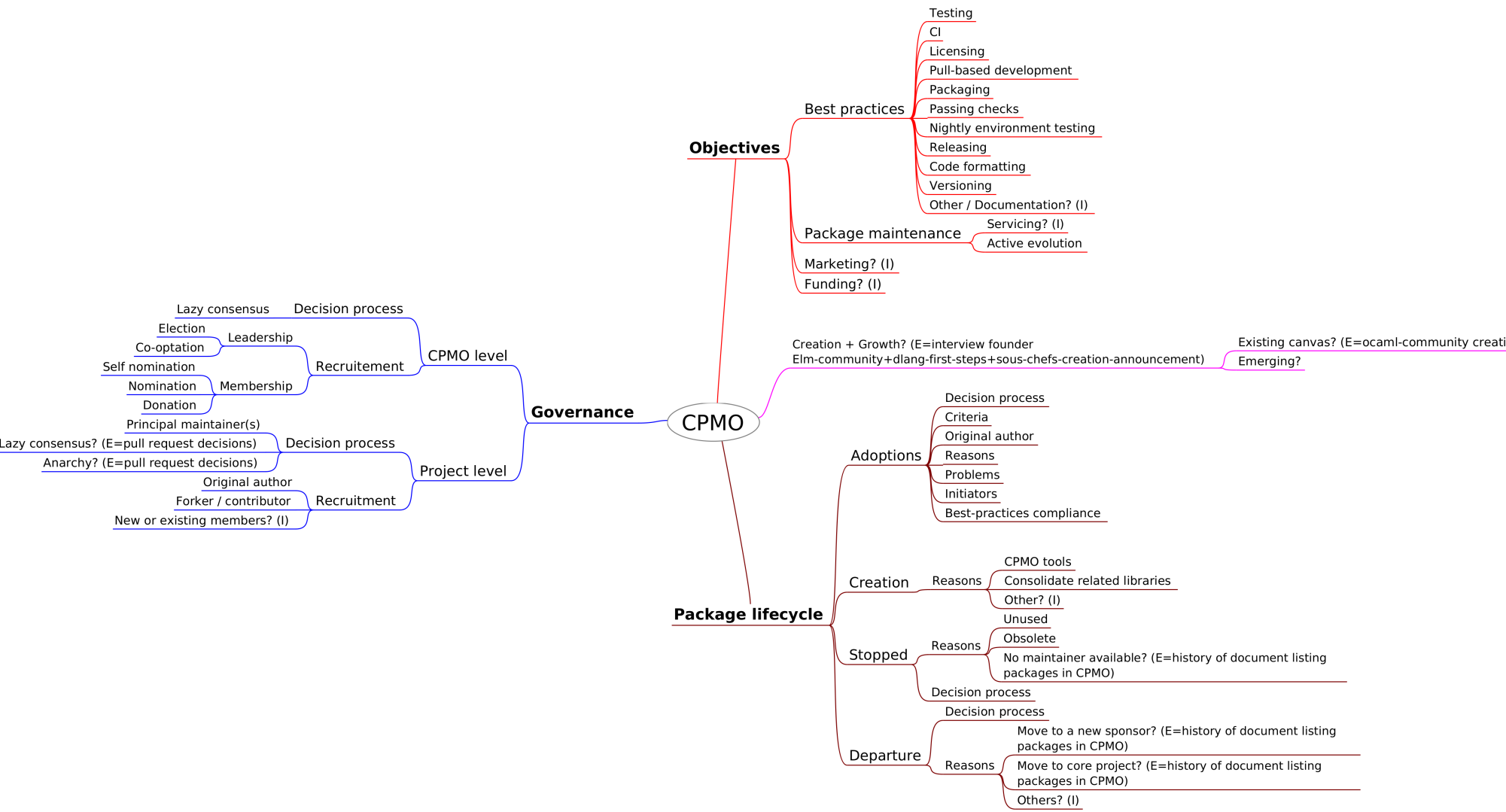




Our study

- We use the qualitative methodology **grounded theory** coming from sociology.
- We apply it on **existing documents** of these organizations.
- We are going to complete this with **interviews** of CPMO participants.
- We have published a "registered report" describing our protocol at **ICSME 2021** and we are expecting to submit our final paper to the EMSE journal soon.

Results


CPMO
Objectives
Best practices
Testing
CI
Licensing
Pull-based development
Packaging
Passing checks
Nightly environment testing
Releasing
Code formatting
Versioning
Other / Documentation? (I)
Package maintenance
Servicing? (I)
Active evolution
Marketing? (I)
Funding? (I)
Creation + Growth? (E=interview founder
Elm-community+dlang-first-steps+sous-chefs-creation-announcement)
Existing canvas? (E=ocaml-community creati
Emerging?
Governance
CPMO level
Decision process
Lazy consensus
Recruitement
Leadership
Election
Co-optation
Membership
Self nomination
Nomination
Donation
Project level
Decision process
Principal maintainer(s)
Lazy consensus? (E=pull request decisions)
Anarchy? (E=pull request decisions)
Recruitment
Original author
Forker / contributor
New or existing members? (I)
Package lifecycle
Adoptions
Decision process
Criteria
Original author
Reasons
Problems
Initiators
Best-practices compliance
Creation
Reasons
CPMO tools
Consolidate related libraries
Other? (I)
Stopped
Reasons
Unused
Obsolete
No maintainer available? (E=history of document listing packages in CPMO)
Decision process
Departure
Decision process
Reasons
Move to a new sponsor? (E=history of document listing packages in CPMO)
Move to core project? (E=history of document listing packages in CPMO)
Others? (I)

1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels
2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations
3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels
2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations
3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

coqbot: a project-specific multi-task bot

- Bot that I created in 2018 and that I have maintained since.
- Helps Coq developers in **everyday tasks** (CI, issue and pull request management, release management, etc.).
- **Contributions** from several Coq contributors: Ali Caglayan, Jason Gross, Pierre-Marie Pédrot, etc. and my intern from 2020 Julien Coolen.
- Paper accepted in the **IEEE Software** special issue on "Bots in Software Engineering".

New features

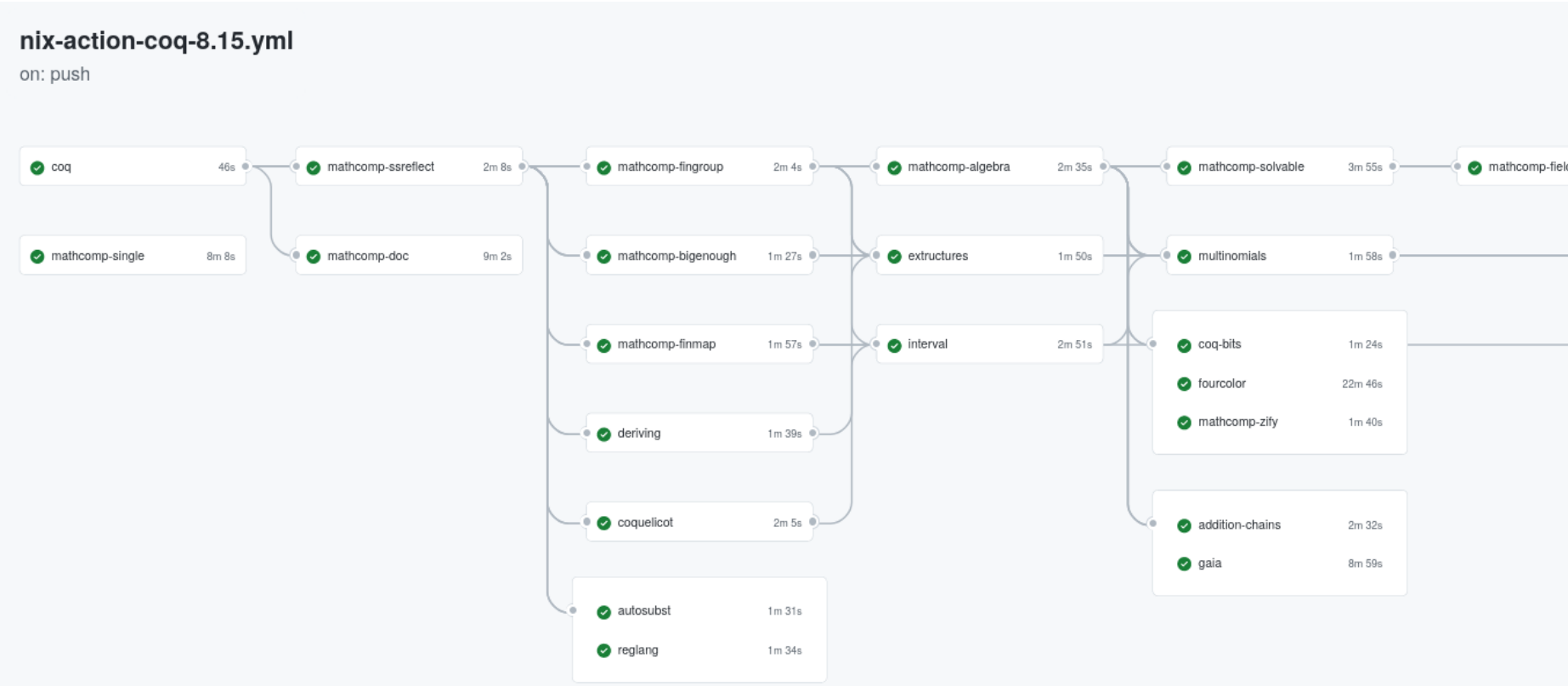
- With Ali Caglayan: integration of the bench infrastructure.
- With Jason Gross: auto-minimization of CI failures.
- With Pierre-Marie Pédrot: auto-closing of stale pull request.

1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels
2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations
3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

The Coq Nix Toolbox

- Developed with Cyril Cohen in 2021.
- Based on the **Nix package manager** (an alternative to opam with functional features).
- Features:
  - **Reproducible** environments.
  - Shared **binary caches** (avoid rebuilding the same thing twice).
  - Support testing with **multiple versions** of dependencies, including development versions.
  - Support **reverse dependency** compatibility testing.
  - Generate GitHub workflows (CI).
- Presented at the **Coq workshop 2021**.

Workflow testing reverse dependencies



1. Making the Coq ecosystem more accessible
   - The Coq Platform: facilitating installation of Coq with packages
   - Hydras & Co.: experiments with new forms of Coq documentation
   - Update on Coq's discussion channels
2. Community organization and management
   - The Coq Community Survey 2022
   - Research on Community Package Maintenance Organizations
3. Making the evolution of Coq and its ecoystem more robust
   - Improvements to Coq's project-specific multi-task bot
   - The Coq Nix Toolbox: supporting advanced CI for Coq projects
   - Auto-minimizing reverse dependency compatibility failures

Reverse dependency compatibility testing

- In Coq, since 2017, all pull requests are **evaluated for compatibility** on a large set of external projects.
- When a project breaks:
  - Impact considered **too big** $\rightarrow$ the change is **abandonned**.
  - Impact deemed **reasonable** $\rightarrow$ the impacted projects are **fixed**.

Integration of the bug minimizer

- Bug minimizer developed since 2015 by Jason Gross.
- Integration in coqbot:
  - Detects when an external project breaks but not the test suite.
  - Proposes to produce a **minimal example**:
    - **reproducing the problem** with the pull request,
    - without breaking in master.
- Has:
  - allowed **evaluating the minimizer** on real examples,
  - helped Coq developers **understand compatibility issues** and enhance the test suite.
- Work published at **ITP 2022**.