2023 Summer of Nix program report

Written by Valentin Gagarin (NGI project manager for the NixOS Foundation)/ Reviewed by Shahar "Dawn" Or (Project organiser for Summer of Nix 2023)

Summer of Nix is a coordinated effort to support free and open source software (FOSS) projects that are funded by the European Commission's Next Generation Internet (NGI) initiative through the NGI0 consortium coordinated by the NLnet Foundation.

The program's main objective is to make more software available as Nix packages or NixOS service modules. This is in order to make the projects developed under NGI0 easier to obtain and run – something that is not a given due to the complexities involved in software development – and thus help end users and developers to reap the benefits of the public funding campaign. Nix itself is a proven technology for highly repeatable software builds and deployments, and supports an ecosystem of tools for building a fully transparent software supply chain.

This year's goals were focused on software developers as users of projects funded through NGI0: Promoting development of the open hardware ecosystem, fostering self-hosting and service portability, and enabling the spread of federated services. Additionally, supported open source projects should be made more visible and easier to discover through a monorepo called NGIpkgs. The new monorepo is modeled after Nixpkgs, currently the the largest, most up-to-date software repository in the world.

The 2023 Summer of Nix program followed the original idea from 2021 in spirit: "work, learn, and meet". It was the first time we experimented with more centralised direction rather than primarily self-organised work. Intermediate status updates were posted in a NixOS Discourse thread.

Summary

- 17 people involved
  - 7 Europe
  - 4 North, Central, or South America
  - 3 Middle East or Central Asia
  - 3 East Asia or Pacific
- 93 000 EUR allocated
- 82 324 EUR spent
  - 26 941 EUR for NGI0 Review
  - 55 383 EUR for NGI0 Entrust
- Activities from June 2023 to December 2023
  - Time worked: 3225.5 h
  - Average compensation: 25.52 EUR/h
  - For comparison, this is roughly the amount of time 3 full-time engineers would spend in the same period, at less than half the cost.

During that time, we had 15 software developers go through a crash course in complexity management with Nix.

- NGI0 Review
  - Got 11 support requests
  - 3 were not actionable, 1 came in very late in the year
  - Handled 7 requests
    - 5 finished, notably
      - Rosenpass
      - librecast
      - openXC7
    - 1 blocked on upstream progress
    - 1 too large to finish
      - GNU Taler
- NGI0 Entrust
  - 5 large projects packaged, including service definitions or plugins
    - KiKit
    - mCaptcha
    - Flarum
    - Pretalx
      - Unfortunately this duplicated work done in Nixpkgs
    - Kbin
  - 1 blocked on upstream progress
  - 4 existing packages updated and migrated into the monorepo
  - dozens stale projects archived
- Monorepo
  - Set up continuous integration
  - Added support for testing with NixOS virtual machines
- Outreach
  - Participated in a podcast recording of Mob Mentality
  - Recorded 12 episodes of 2023 Nix Developer Dialogues
  - Informed more than 60 companies which use Nix of the program and its graduates

Evaluation

We can claim that developer experience around using or contributing to software projects supported through NGI0 was substantially improved. Notable progress was made in all areas of focus for this program.

We did not reach the goal of setting up a convenient presentation layer in NGIpkgs. This was for scheduling reasons; the allocated budget is still available.

The program was accompanied by a public narrative that reached on the order of 2000 people (with view counts between 300 and 4400). This corresponds to the known number of active participants in the Nix community and adjacent groups.

- What went well:
  - Knowledge sharing: Mob programming is very effective
  - Code quality: Mobs worked very diligently; packages and services are tested automatically
  - Collaboration: Participants and project authors successfully worked together
  - Focus: The teams solved a handful of hard problems
  - Task curation: Everyone knew what to work on and why
  - Monorepo: Reduced overhead and improved developer experience
- What should be improved:
  - Diversity: No non-male participants this time
  - Scope: The delivery goals were ambitious and task complexity hard to predict
  - Cohesion: The effort got diffuse towards the end of the year, not all work got done; more check-ins needed
  - Nixpkgs tooling and documentation: Still a major source of friction
  - Efficiency: Mob programming is not a silver bullet; we have to figure out a sensible balance which tasks to do alone or in a group, and when

Overall, and as predicted, better preparation greatly reduced the challenges reported from the 2021 and 2022 programs — such as clear objectives and task assignments — while keeping some freedom for participants to follow their tech interests. This came at the cost of a smaller number of software projects that were packaged in this period.

Lessons learned

Many insights from the 2021 and 2022 editions were implemented successfully, and prior improvements could be reproduced. Some things have not been addressed though, but should be in the next iteration:

- Improve the technical introduction so participants can get started more quickly.

  For example, provide pointers to documentation on key aspects of the ecosystem, or re-introduce the mentor role.

- Keep up motivation until the end.

  A final presentation or closing ceremony could help. Participants should get NixCon 2024 tickets for an opportunity to meet developers in person. This could be combined with a hiring event. This time we had two participants present at a Nix workshop, which was well-received by the audience.

- Consider more support roles, such as technical writer, community manager, web master.

This time we focused exclusively on software development. In 2022 there were alternative tracks, but those did not see much interest, possibly because they were announced after the application phase. But its clear that eventually, if we want to improve the overall user and developer experience, we will need dedicated work done in various problem domains. This is also a chance to increase diversity among participants.

- Plan for additional outreach efforts.

  Make time for participants to write experience reports, technical articles, or other content to convey what we're doing and why. This could also be implemented as a follow-up stipend.

- Make Summer of Nix more than just about NGI.

  Asking companies to submit and fund project proposals could help to leverage the visibility of the program, and to make progress on long-standing issues that can't be addressed through the purpose-bound grants. It could also be coupled with a hiring event to help companies with talent-acquisition.

We also learned some new lessons:

It's vital to have at least one person to be available for answering questions and making more high-level decisions. That generally went well this time, but would have benefited from more consistency and a more regular schedule.

The past two Summers of Nix produced impressive numbers of new packages. This year's experience shows a more realistic price of getting things to work smoothly end-to-end, an effort that includes developers building up the required contextual knowledge. Actually finishing things is sometimes surprisingly expensive! In particular those high-value targets with far reach or great potential, which we originally aimed at, tend to be beasts of complexity.

Taking note of that was only possible due to more detailed accounting than was done in prior years. That also incurred additional busywork, and once again time sheets and invoicing raised repeated questions by participants. Next time we will set up a centralised form for reporting, and refine documentation and contracts.

More generally, over time we have accumulated a lot of information in many different places, which get increasingly hard to navigate. The monorepo is a step in the right direction for consumers. This should continue by archiving repositories created in the past years. Something similar should be done for internal information that still needs to be unified, published, or deleted. For example, we turned out to actually need only 3 out of 9 Matrix rooms, and none of the 15 existing private repositories. Eventually all information on the concerning program organisation should live in one public repository which documents all related resources and procedures.

Since the program is now supposed to be continuous in order to support ongoing projects funded by NLnet, opportunities to continue work after the Summer of Nix period should be announced in the application phase to make the program more attractive and ensure that more people are available to take on tasks.

Goals for the next years

Having overseen the 2023 program, participated in overcoming technical and organisational challenges, observed the 2021 and 2022 editions, and assisted with their evaluation, I recommend to further narrow the focus on key strategic objectives in the following years. Assuming that a high-level goal of the NGI0 program is to enable viable participatory alternatives to proprietary offerings, we have to set good examples demonstrating competitive quality – and find ways to do this repeatably, independent of particular individuals, and with very limited means since those won't be available after public funding runs out. On the technical side this would mean for future program participants to work on providing great user and contributor experience in terms of configuration and deployment, with selected projects that will have the largest impact on the broader ecosystem, and where this is most likely to succeed within our budget constraints. On the organisational side it would mean to select such projects and find out under which conditions such success is more likely, and our program reports are part of that investigation.

With the current grant agreements in place, there is budget for another three iterations of Summer of Nix. The next periods should align with the explicitly defined NGI0 program goals:

1. Operate the agile, effective and low-threshold funding mechanism of NGI0
2. Attract, recruit and select the best possible projects to the program
3. Provide multidisciplinary technical and strategic support
4. Scale and become self-sustainable, with providing best practices and advice
5. Strong and future proof ecosystem for long-term sustainability

Generally, the low-threshold funding mechanism seems to work. Currently there are more than 150 projects supported through NGI0 that have been packaged since 2021. Many of them have been abandoned, but some are very successful and actively developed. What seems to be lacking, in my opinion, observing from the outside as well as experiencing it on the inside of the program, is an overview of what we consider high value, good quality, successful projects; in short: curation.

Summer of Nix in a sense has become a litmus test for project maturity: Participants have repeatedly taken software for a ride when trying to ensure that it builds and runs repeatably with Nix. This requires stable interfaces, useful documentation, and responsive maintainers. Participants come from diverse technical backgrounds, and their attention and critical inquiries are helpful feedback for project authors. We should continue in that fashion, and in 2024 focus

on showcasing exceptional open source software and ensure there are up-to-date, readily available Nix packages or service modules that can be put to use with minimal effort. This should also involve expanding consumer-facing documentation in NGIpkgs with guides and examples.

What still seems unanswered is the question of sustainability. This is not a technical problem, but rather a social and economic one: Nix infrastructure is designed for and demonstrably capable of ensuring long-term availability of packages, and the Nix ecosystem already has almost all the tooling needed for maintaining sophisticated setups. Primarily lacking are comprehensive reference documentation, a corpus of best practices or recipes for common use cases, and more structured approaches to teaching the underlying principles. Continuing with Summer of Nix, NGIpkgs should eventually complement Nixpkgs as a vital place of exchange and as a curated repository of useful open source software.

What is unclear at the moment is how to enable continued maintenance of NGIpkgs, and the effort required to effectively adopt Nix tooling into project authors' development processes is related to that. With Summer of Nix we successfully manage to onboard participants quickly into the Nix ecosystem, and embedding them into the community keeps growing our network of experts. And while Nix packaging reduces the cost of adoption of and contributions to software, project authors tend to keep relying on external support to evolve their Nix setups. Apart from maturing the NGIpkgs software distribution, in the next years we should therefore focus on expanding the knowledge sharing to project authors and enable them to maintain their own packages. Ideally participants should also, based on their practical experience, contribute documentation and insights back to the Nix ecosystem to make future users and contributors less dependent on availability of individuals.

A possible set of objectives in the next years, apart from concrete deliverables, could therefore be:

- 2024: Select and showcase highlight projects
- 2025: Collaborate with project developers
- 2026: Scaled onboarding and sustainable maintenance