

# Scaling up open source projects

Congratulations! You have good running  
open source project

But... how do I scale?

# What does scaling mean?

Scaling means a project is able to...

- build a diverse and vibrant contributor/maintainer base
- enable 100s or 1000s of developers/users
- support a vendor ecosystem of 100s of products

Question: How can a project do that?

# What does scaling mean?

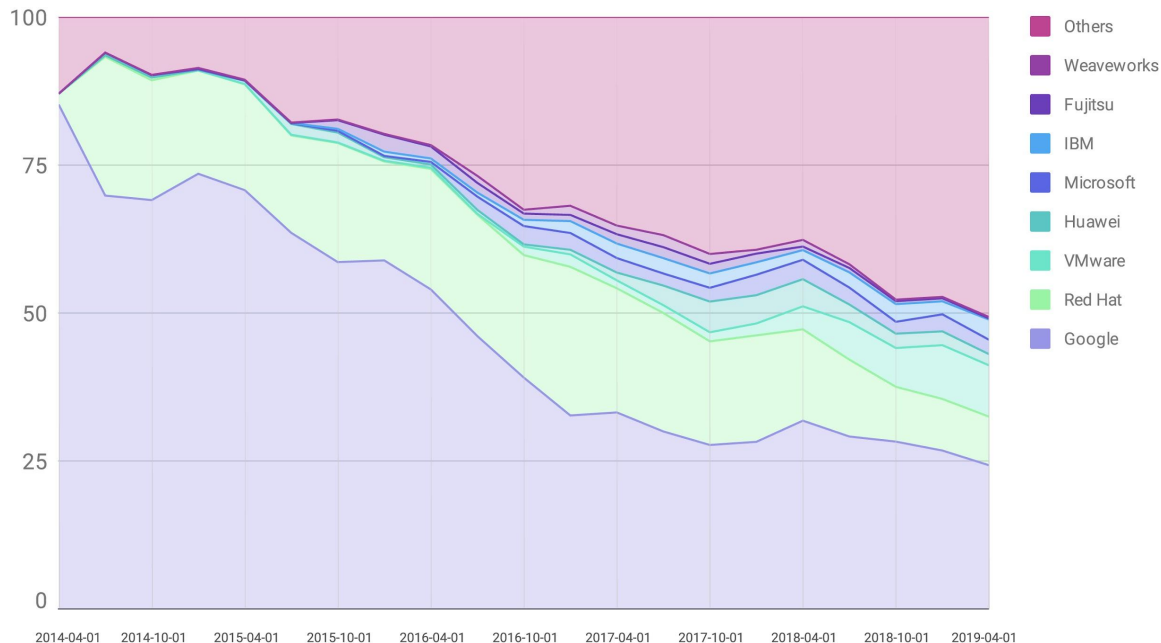
Scaling means a project is able to...

- **build a diverse and vibrant contributor/maintainer base**
- enable 100s or 1000s of developers/users
- support a vendor ecosystem of 100s of products

# What does diversity mean?

- Multiple organizations committing, no one organization more than 40% of committers
- Committers from different races, genders, nationalities, locales, etc

A healthy project is a diverse one. How can a project do this?



Percentage breakdown of contributions by company since Kubernetes project launch

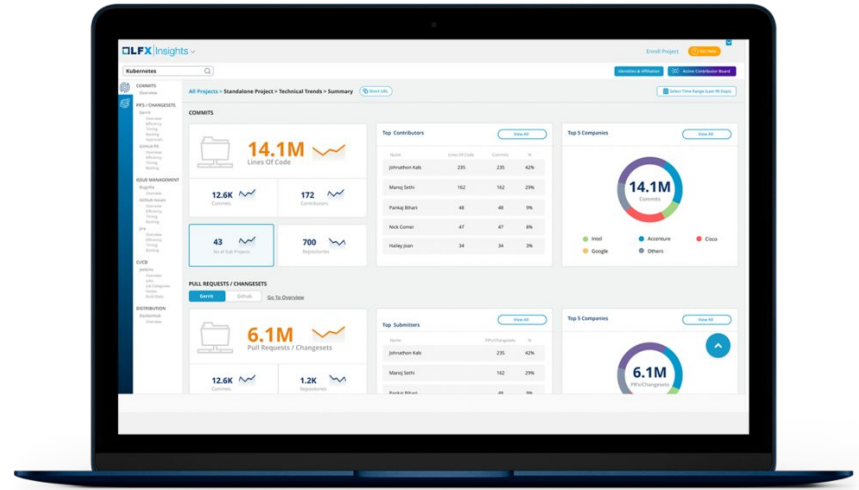
# First step is to measure

Being able to scale means knowing how the project operates. Without data, you don't know where to begin.

Key data points to consider:

- Rate of contributions
- Number and diversity of contributors
- Collaboration and engagement in mailing lists, discussion groups, chat channels.

[LFX Insights](#) is a great tool to dig more into this.



*LFX Insights*

# Build an on-ramp for new contributors



Contributors often aren't sure how to best engage. Open source projects can feel intimidating to them.

Some ideas to make that on-ramp easier.

- Have a clear, concise contributors guide.
- Use the 'good first issue' tag on issues that are good ones for new contributors to tackle.
- Hold regular, open developer communities meetings that encourage potential and new contributors to engage the maintainers.
- Ensure maintainers are engaging potential contributors in a positive, encouraging way in pull requests, issues, and responding in online forums.

## Contributor's Guide

If you're reading this you're probably interested in contributing to [hyper](#). First, I'd like to say: thankyou! Projects like this one live-and-die based on the support they receive from others, and the fact that you're even *considering* supporting [hyper](#) is incredibly generous of you.

This document lays out guidelines and advice for contributing to [hyper](#). If you're thinking of contributing, start by reading this thoroughly and getting a feel for how contributing to the project works. If you've still got questions after reading this, you should go ahead and contact [the author](#): he'll be happy to help.

The guide is split into sections based on the type of contribution you're thinking of making, with a section that covers general guidelines for all contributors.

## All Contributions

### Be Cordial Or Be On Your Way

[hyper](#) has one very important guideline governing all forms of contribution, including things like reporting bugs or requesting features. The guideline is [be cordial or be on your way](#). **All contributions are welcome**, but they come with an implicit social contract: everyone must be treated with respect.

This can be a difficult area to judge, so the maintainer will enforce the following policy. If any contributor acts rudely or aggressively towards any other contributor, **regardless of whether they perceive themselves to be acting in retaliation for an earlier breach of this guideline**, they will be

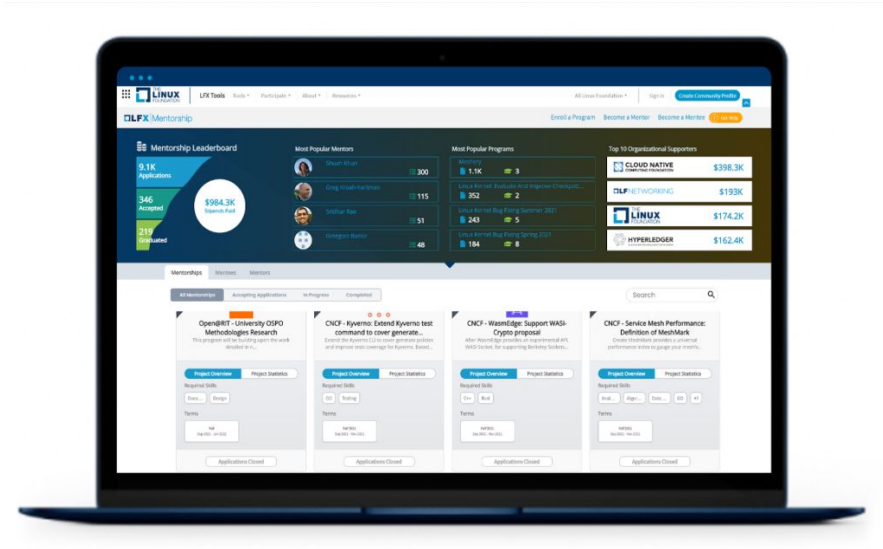
*Hyper Contributor's Guide*



# Mentorships

Offering paid mentorships helps draw new contributors into the project tackle key issues. Mentorships are also a great onramp for new to industry developers to learn more and explore career opportunities.

- [LFX Mentorship](#) is a great place to run project funded mentorships
- If your project doesn't have much funding, explore programs like [Google Summer of Code](#) and [Outreachy](#)



*LFX Mentorship*

# Recognize great contributors

Open source contributors and maintainers often feel underappreciated for the work they do. Additionally, contributors use the work they do in open source as a “portfolio” which helps grow their career.

- Recognize contributors in release notes to formally thank them.
- Develop a badging program using a tool like [credly](#) to let contributors showcase their community status.
- Give them some swag! Leverage tool like [Spreadshop](#) and give them some codes for free merchandise.

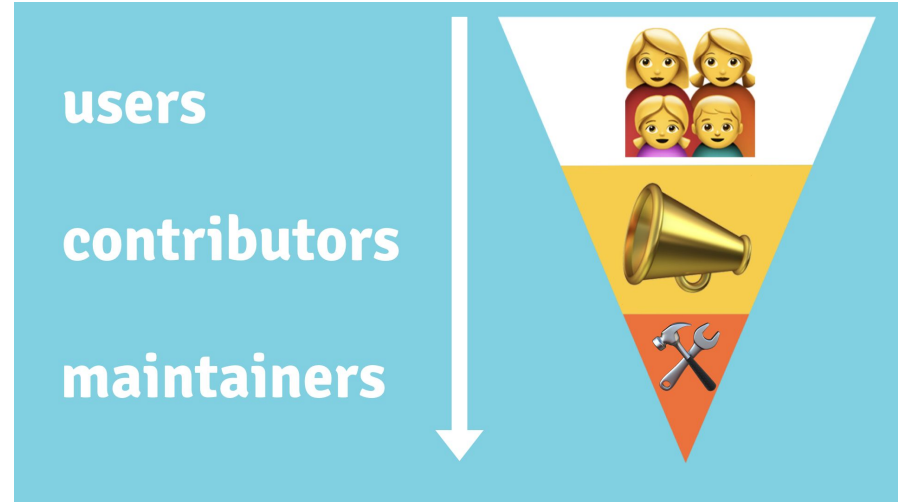


# Communities have a lifecycle too

Sustainable open source projects aim to train the contributors of today to be the maintainers of tomorrow. With that, you need space for these new maintainers to step in for the maintainers of today.

Preparation for a project for this cycle should include

- Create clear documentation on processes and roles for things like releases, code contributions, and general decision making.
- Let new maintainers pair with existing maintainers to mentor them in the role
- Make sure current maintainers can step away from the project with minimal negative impacts.



Source: [The Open Source Contributor Funnel \(or: Why People Don't Contribute To Your Open Source Project\) | Mike McQuaid](#)

# What does scaling mean?

Scaling means a project is able to...

- build a diverse and vibrant contributor/maintainer base
- **enable 100s or 1000s of developers/users**
- support a vendor ecosystem of 100s of products

# What are the challenges someone wanting to use the code might have?

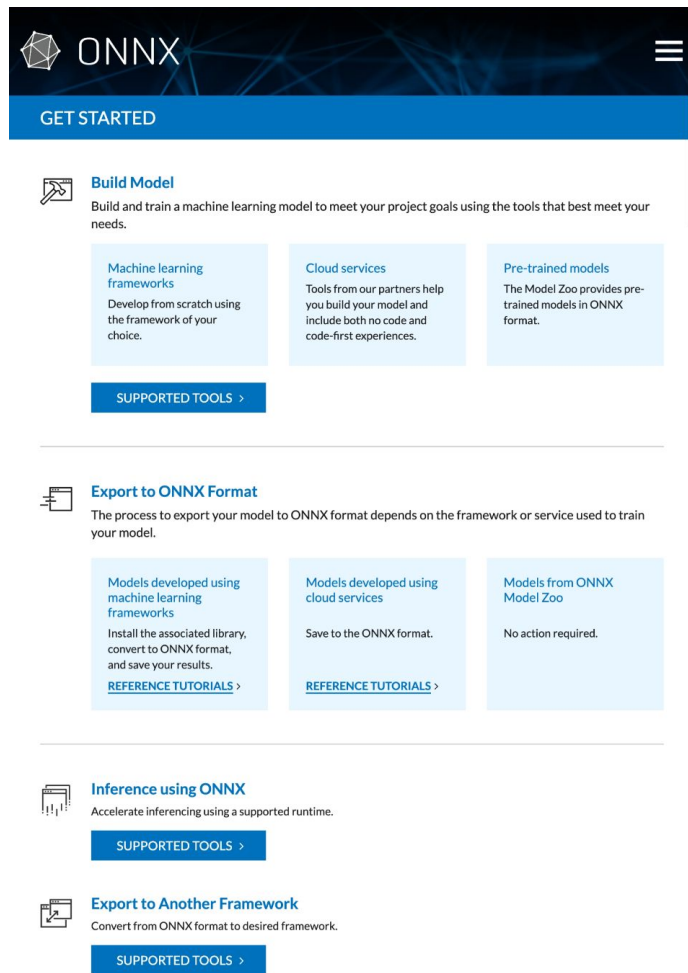


- How can I get started quickly?
- Where can I find some tips and tricks on common use cases?
- How can I showcase my competence ( and leverage that to build my career )?

# Getting Started guides

Helping developers get up and running quickly is the most important indicator of success.

- Make the instructions concise and easy to understand.
- Use videos to help
- Point developers to other resources to help them on the next steps.



The screenshot shows the 'GET STARTED' section of the ONNX website. At the top, the ONNX logo is on the left and a hamburger menu icon is on the right. Below the logo is a blue navigation bar with the text 'GET STARTED'. The main content area is divided into several sections, each with an icon and a title. The first section is 'Build Model', which includes three sub-sections: 'Machine learning frameworks', 'Cloud services', and 'Pre-trained models'. The second section is 'Export to ONNX Format', which includes three sub-sections: 'Models developed using machine learning frameworks', 'Models developed using cloud services', and 'Models from ONNX Model Zoo'. The third section is 'Inference using ONNX', and the fourth is 'Export to Another Framework'. Each section has a 'SUPPORTED TOOLS >' button.

**ONNX**

GET STARTED

**Build Model**  
Build and train a machine learning model to meet your project goals using the tools that best meet your needs.

**Machine learning frameworks**  
Develop from scratch using the framework of your choice.

**Cloud services**  
Tools from our partners help you build your model and include both no code and code-first experiences.

**Pre-trained models**  
The Model Zoo provides pre-trained models in ONNX format.

SUPPORTED TOOLS >

**Export to ONNX Format**  
The process to export your model to ONNX format depends on the framework or service used to train your model.

**Models developed using machine learning frameworks**  
Install the associated library, convert to ONNX format, and save your results.  
REFERENCE TUTORIALS >

**Models developed using cloud services**  
Save to the ONNX format.  
REFERENCE TUTORIALS >

**Models from ONNX Model Zoo**  
No action required.

**Inference using ONNX**  
Accelerate inferring using a supported runtime.  
SUPPORTED TOOLS >

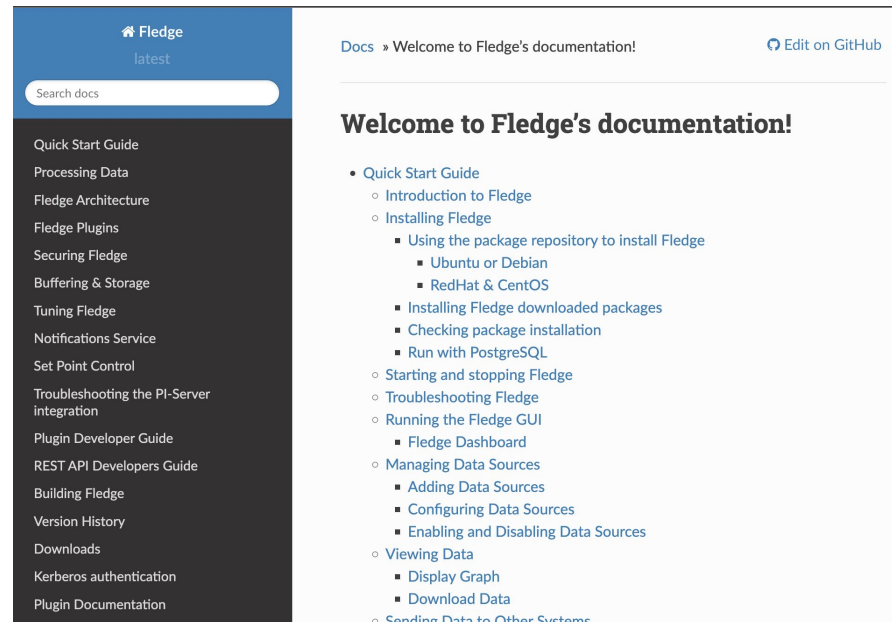
**Export to Another Framework**  
Convert from ONNX format to desired framework.  
SUPPORTED TOOLS >

# Solid documentation

Good documentation unblocks developers and users to reduce frustration and become productive quickly.

Writing great technical documentation is a unique skill - consider bringing in someone.

- Hire a technical writer. A great resource for finding writers is [Write the Docs](#).
- Leverage a program such as [Google Season of Docs](#).
- Use a tool like Read the Docs to as a framework for making documentation easy to read and navigate.
- Not everyone is a native English speaker. Consider internationalization for the documentation.



The screenshot shows the Fledge project documentation website. The header includes the Fledge logo and 'latest' version. A search bar is present. The left sidebar lists various documentation topics such as 'Quick Start Guide', 'Processing Data', 'Fledge Architecture', 'Fledge Plugins', 'Securing Fledge', 'Buffering & Storage', 'Tuning Fledge', 'Notifications Service', 'Set Point Control', 'Troubleshooting the PI-Server integration', 'Plugin Developer Guide', 'REST API Developers Guide', 'Building Fledge', 'Version History', 'Downloads', 'Kerberos authentication', and 'Plugin Documentation'. The main content area displays 'Welcome to Fledge's documentation!' with a link to 'Edit on GitHub'. Below this is a table of contents for the 'Quick Start Guide' section, listing items like 'Introduction to Fledge', 'Installing Fledge' (with sub-items for package repositories and installation), 'Starting and stopping Fledge', 'Troubleshooting Fledge', 'Running the Fledge GUI', 'Managing Data Sources', 'Viewing Data', and 'Sending Data to Other Systems'.

*Fledge project documentation*

# Outreach

There are millions of open source projects. Projects need to work to stand out.

- Write blog articles on a regular basis that outline development milestones and new feature/functionality.
- Create videos and share via social media of how to get started, sample use-cases.
- Present at events that are pertinent to your project.
- Do meetups and hackathons where potential contributors are.



[Wikimedia Hackathon 2015](#)



# Training and Certification

Organizations that want to use your project need to know there is talent out there to help them out. They also need to enable their own staff.

- Develop formal training programs that cover the key aspects of using the project.
- Build a certification program that lets individuals showcase their skills in the technology.

[LF Training](#) is a great partner in developing training and certification programs.



# What does scaling mean?

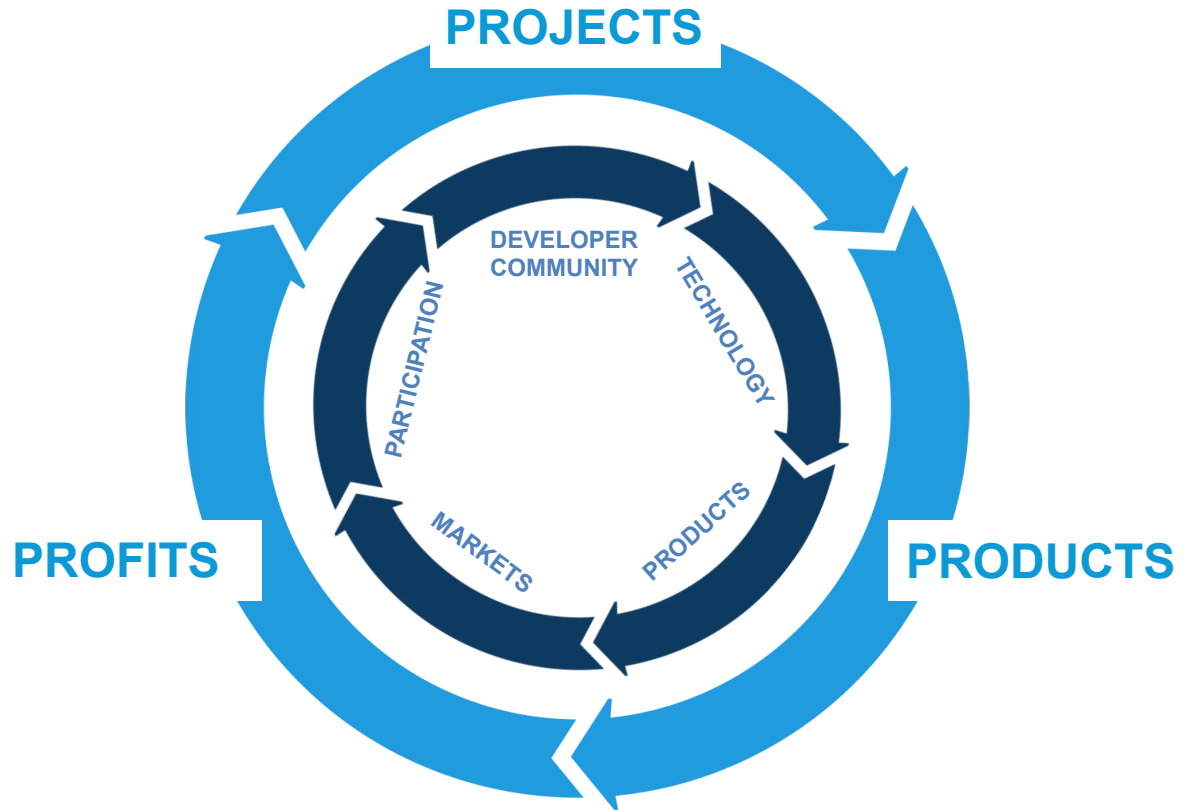
Scaling means a project is able to...

- build a diverse and vibrant contributor/maintainer base
- enable 100s or 1000s of developers/users
- **support a vendor ecosystem of 100s of products**

# Productization is part of the cycle of a sustainable project

Successful projects depend on members, developers, standards and infrastructure to develop products that the market will adopt.

Helping define that ecosystem makes it easier to bring products to market and end-users to adopt.



# Case Studies and User Stories

End-users look to other end-user experiences to help shape their decision making.

Building a library of case studies and/or user stories showcases end-user use-cases, giving confidence to others of successful use of the project and business impacts.

[LFN User Stories](#)



## DT Deploys ONAP In O-RAN Town




Deutsche Telekom is on their way of bringing ONAP from the labs into pilot production. In their O-RAN Town project, DT deployed in the city of Neubrandenburg a multi-vendor Open RAN trial network for 4G and 5G services with massive MIMO integrated into the live network — the first in Europe. To automate services on all network domains, DT introduced a vendor-independent Service Management and Orchestration (SMO) component based on ONAP open source. The SMO is to be at the heart of complete lifecycle management of all O-RAN components in this deployment. Learn from Marc Fiedler, Sebastian Zechlin, and Andreas Geissler from Deutsche Telekom about how they overcame deployment challenges and the benefits using ONAP.

### Learn More:

[Download 1-Page Overview \(PDF\)](#)

Webinar: (Nov 18, 2021) [\(Video\)](#) [\(Slides\)](#)

## More User Stories

 <a href="#">User Story</a> <b>Verizon</b> OpenDaylight is Verizon's Directional SDN Controller <a href="#">READ STORY</a>	 <a href="#">User Story</a> <b>Orange</b> Orange Deploys ONAP in Production <a href="#">READ STORY</a>	 <a href="#">User Story</a> <b>Spark</b> Spark Embarks on Telco Cloud Automation Journey with ONAP <a href="#">READ STORY</a>
---	--	---

# Build a conformance program

Conformance programs enable the project to showcase an ecosystem of products and services leveraging the project. It also lets project ensure that there is interoperability between vendor solutions which gives end-users choice.

- Technical requirements defined by the community
- Program is administered by the project staff to ensure non-bias
- Vendors get use of exclusive marks and listing in a solution directory with equal footing with other solutions
- Projects may choose to provide additional benefits as they make sense for the ecosystem.



*Examples of Conformance Programs with other Linux Foundation projects ( [Zowe](#) and [Kubernetes](#) )*

## Read more on project success



- [Case Studies - Linux Foundation](#)
- [CNCF Project Journey Reports](#)

Thank You!