Much of the software that powers the world's largest companies, protects

our personal data, or encrypts national security information is open to

the public. Anyone can download the source code behind Facebook's user

interface, Google's Android operating system or even Goldman Sachs' data

modeling program, and use it as a building block for a totally new

project. What's more, lots of this software is actually developed

collaboratively, created and maintained by an army of thousands, from

unpaid volunteers to employees at competing tech companies.

As a kid in a small town in Virginia, I could get connected to the best

developers anywhere on earth and learn from them and even read the code

that they'd written. I really wanted to give back and these people were

always my heroes, so I wanted to participate too.

And my voice mattered, it was just immediately, I was hooked.

This is the collaborative world of open-source software, where code is

written and shared freely.

If individuals catch a bug or see an opportunity for improvement, they can

suggest changes to the code and thereby become a contributor to some of

the biggest software projects on Earth.

But this model hasn't always been the norm.

At the dawn of the Internet era through the late 1990s, proprietary

software proliferated.

Microsoft even went so far as to call open-source un-American and bad for

intellectual property rights.

Software was a finite commodity that people hoarded and wanted to sell as

a product. Open-source software was only developed and maintained by a

dedicated few. And there was this fringe world, there was this academic

world who were creating software according to their own rules and sharing

it publicly and making it free.

Certainly dreamed, like wouldn't it be awesome if we could sort of take

over the world? Now, open-source has, essentially, taken over the world.

Companies in every industry from Walmart to ExxonMobil to Verizon have

open-sourced their projects.

Microsoft has completely changed its point of view and is now seen as a

leader in the space.

And in 2016, the U.S.

government even promised to open-source at least 20 percent of all its

new, custom developed code.

So whether you know it or not, you are relying on the volunteer labor in

many cases of thousands of strangers from around the world.

In the 1970s, the M.I.T

Artificial Intelligence Lab had a printer that regularly jammed.

So staff programmer, Richard Stallman, altered its source code, so that

when problems arose it would send a message to everyone in the lab saying,

"Go fix the printer." When the lab finally got a new printer, Stallman

discovered its source code was inaccessible.

He asked for the code, got refused, got upset and ultimately quit his job

to develop a completely open operating system called GNU, in 1983.

With this, Stallman spearheaded the free software movement from which the

open-source movement was born.

It's sort of a very natural way to work together.

Collectively, if everybody comes and contributes their piece, you end up

with something that's a lot greater than something that an individual

could contribute on their own. But throughout the 1980s and the 1990s,

proprietary software is still dominated.

And that was a very lucrative way of producing and selling software, and

created an incentive for large technology companies to create a

proprietary de facto standard.

It was against this backdrop that the open-source operating system Linux

was unceremoniously released in 1991.

It incorporated many elements from Stallman's GNU project, but was mainly

used by hobbyists looking for an alternative to Windows or MacOS.

Throughout the decade though, Linux gained momentum as large companies

took advantage of its flexibility and tweaked the software to their

specific needs. By the turn of the century, NASA, Dell and IBM were all

using it. The platform itself changes nine times an hour.

Ten thousand lines of code are added to Linux every day.

About five thousand lines are changed and about eight thousand lines are

removed. It is by far the highest velocity, most effective software

development process in the history of computing.

As Linux grew, other open-source projects were also gaining popularity,

like the database management system MySQL, the Perl programming language

and the web server Apache.

But for the layperson at the turn of the century, the rise of these

technologies could have gone unnoticed.

After all, hardly anyone ran Linux on their personal computers.

But then in 2008, Google released Android devices, which run on a modified

version of Linux. Suddenly, the operating system blew up the smartphone

market. We are still overwhelmed with the amount of innovation that is

happening in that ecosystem of Android.

Today, there are over 2.5

billion active devices using Android.

As Google demonstrated, businesses were increasingly relying upon this

complex web of open-source technologies to build products and platforms

quickly. And whether they knew it or not, this also meant that they were

depending upon the vast open-source community to maintain this software.

The ability for one company to produce the amount of software that's

required for any modern technology, product or service became

overwhelming. Today, in a modern luxury automobile, there are more lines

of software code than in an F-15 fighter jet.

There's just simply too much software to be written for any single

organization to write it themselves.

Ninety nine percent of Fortune 500 companies use open-source.

Every web server is pretty much Linux.

Most people choose to use open-source programming languages.

It's this like amazing buffet.

You just come in and take this, this, this, this and you slap together

something. You can build, you know, amazingly powerful products with very

little work. The sheer increase in volume drove a need amongst developers

for a central repository where they could collaborate on these huge

projects. And in 2008, GitHub provided an answer.

What GitHub really did for open-source is it kind of standardized the way

that people can contribute to open-source projects and interface with

them. And so that any developer, anywhere on earth knew how to contribute

to a project on GitHub.

And that fed this explosion of open-source activity.

Today, Github hosts the grand majority of the world's open-source software

projects. And in 2018, Microsoft even acquired GitHub, affirming the tech

giant's commitment to open-source development.

We have about 140 million open-source or software projects that are on

GitHub, and over 50 million of those have been added just in the last

year, so it's growing incredibly fast.

And the community has expanded far beyond idealistic hobbyists, as major

companies are increasingly leading the charge, spearheading the

development of open-source projects in-house.

So we see companies like Intel, Google, Facebook, Amazon, Microsoft all

contributing heavily to open-source, Googlers have been contributing to

over 28 thousand projects in 2018.

This number includes projects that Google has driven as well as

contributions it's made to projects led by other companies or individuals.

In the open-source world, you have these fierce commercial rivals who

collaborate every day together.

And they haven't signed anything.

And it's not just software companies.

ExxonMobil has open-sourced its developer toolkit.

Walmart open-sourced its cloud management platform, and Goldman Sachs

recently open-sourced its data modeling program.

This open-source way of working turned out to be better.

Because even in a big tech company where you might have 20 or 30 or 50

thousand developers, you can't compete with the 40 million developers that

are now on GitHub working on open-source every day.

Put simply, open-source development has become the new norm.

And so I expect to see more and more investment into open-source

communities and more and more projects as well as companies finding ways

to commercially monetize the activity that's collected across all of these

different open-source communities.

So how is open-source monetized, when the product is basically given away

for free? Basically, the answer lies in selling support services,

subscriptions and/or commercial versions of the software.

Red Hat, founded in 1993, was the first to figure out a successful

business model, which relies upon selling support services for its

operating system, Red Hat Enterprise Linux.

Anyone can download the software for free, but if businesses want

technical support and greater security, they'll need to buy a

subscription. After decades of commercial success, IBM officially acquired

Red Hat in 2019 for 34 billion dollars.

It was the largest software acquisition in history.

It's really groundbreaking, and it just shows the power and success of

open-source across the world, across the industries.

While Red Hat's 100 percent open-source model has been hard to replicate,

other companies like database program MongoDB and integration platform

MuleSoft, rely upon an "open-core" model, meaning the basic features are

free, but add-ons and other useful elements are proprietary.

As these companies have racked up multi-billion dollar valuations, there's

no doubt that on the enterprise level, there's big money in open-source.

But as for the individual developers, the hobbyists who contribute to and

maintain open-source projects just for fun, their path to profitability is

much less clear. There are a lot of different models for how people make

money in open-source.

One of the models is actually they don't.

A lot of them are volunteers and they do this in their free time, in the

evenings and on the weekends.

But sometimes these "just for fun" side projects end up becoming widely

used, critical to the internet infrastructure that we generally take for

granted. Certain projects get so popular and so widely used that, you

know, they're in every product.

They're used by every company. And then this result of like, you know,

massive Fortune 100, Fortune 500 companies building their businesses on

top of this code that's written by hobbyists for who knows what reason,

right? As one might imagine, problems can arise when critical systems are

based on software that's maintained by unpaid volunteers with no

professional obligation to see to the maintenance of the project.

This issue came to a head in 2014, when the security vulnerability dubbed

Heartbleed was found in OpenSSL, an open-source encryption technology

that's used by the majority of web servers to protect user's personal

data. The flaw has gone undetected for about two years and has exposed

millions of usernames, passwords and possibly credit card info as well.

They left this lock capable of being picked, because they didn't write the

code quite right.

And when they looked into it, the OpenSSL team was tiny.

It was just a few people, who were mostly working on donations, and their

donations had started to dry up.

And understandably, these incredibly talented programmers had a hard time

justifying spending full-time on this, even though it was one of the most

important building blocks of the entire Internet.

Ultimately, organizations like the Linux Foundation pulled together to

provide financial support for OpenSSL, as well as other critical pieces of

underfunded open-source software.

But the disaster served as a wakeup call for an industry that still

largely relies upon unpaid labor.

The burden on the maintainer can become quite intense.

You have sort of all these businesses coming and saying like, you know,

your code is broken, and it's preventing me from getting my job done.

And so you get this like almost intense guilt, I feel like, a lot of

maintainers have a guilt.

Now, companies are taking note and helping to formalize new funding

models. In 2019, GitHub rolled out their Sponsors program, which allows

developers to give and receive recurring donations for their work.

We already have some people who are making their full living on GitHub

Sponsors, so they don't have another job.

And so that's huge for us.

And that's really the key number that we are trying to move up.

For developers like Aboukhadijeh, making a full-time living around

open-source is indeed the dream.

I prefer the freedom of being able to just follow my interests wherever

they go and solve whatever problems I think are interesting.

I'll never make money from some of the stuff I've worked on, but I think

I've done a lot of good in the world by working on it.

It's still early days, but if funding models like GitHub Sponsors pay off,

we may see a new class of software engineers eschewing traditional tech

jobs in favor of independent, open-source work.

Open-source has created so much technological progress.

But because it never quite solved this funding problem in those early

years, I think that we are at a point now where we could lay the

groundwork for the next stage of open-source.

This could also mean expanding the cooperative ethos to industries outside

of software. As the idea of sharing technology and collaborating

collectively expands, we're moving into open hardware initiatives,

data-sharing initiatives, and that's really going to be the future.

After all, the success of open-source reveals that collaboration and

knowledge sharing are more than just feel-good buzzwords, they're an

effective business strategy.

And if we're going to solve some of the world's biggest problems, many

believe that we can't afford to hoard our resources and learnings.

We have environmental concerns.

We're trying to understand things like cancer and Alzheimer's.

These are major collaborative efforts.

The complexity of building these technologies isn't going down, it's only

going up. We can get that technology out there faster when everybody works

together.