



**How to create a
successful open
source project.
Hangzhou, China, 2013**

**Michael Widenius
MySQL & MariaDB founder
monty@mariadb.org**

Erich S. Raymond & Linus Torvalds

"Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix will be obvious to someone"

or how [Linus Thorvalds](#) rephrases this

"Given enough eyeballs, all bugs are shallow"

This talk is for

- People wanting to understand how open source works.
- People wanting to create an open source project.
- People wanting to participate, drive or fork an open source project.

In addition I will talk about

- How to make money with Open Source
- Difference in business model between MySQL and MariaDB.

Top reasons for creating an open source project

- Solving one of ones personal problems
 - > For the company you are working (MySQL/PHP)
 - > As a research project, part of your hobby (LINUX)
- Re-license an existing closed source project to open source to get the benefits of open source (Netscape)
- Wanting to earn money and at the same time do something good.
- Wanting to give back something to the open source community.

Developing software open source is in human nature

- You solve your own problems and get free help and development efforts from others while doing it.
- You participate to increase your reputation (to get a new job?)
- You use open source because it's less expensive (and re-usable)
- You give patches back to not have to maintain them.

There are of course people that participate because they believe in the open source cause or want to help others, but these are a minority (and most still gets paid for this).

Before starting a new open source project

- Check that if there is already an existing actively developed project.
 - > It's always better to participate than to do new project or a fork!
- If there are only old dead projects, do some research why they failed and learn from their mistakes.
 - > Sourceforge, Github and Launchpad is filled with dead projects.
- Find a company or a group of users that wants to work with you to define the scope of the project.
 - > You want to have users ASAP that are using the project in production!

It's not just software

- The most important part with open source is to create an active community!
- You also need (to interact with the community):
 - > Web pages (someone needs to design these)
 - > A forum or a knowledgebase, email lists, bug system.
 - > Documentation & localization
 - > Packages, build system, mirrors (for downloads)
 - > Source code repository

In the beginning you can use github or launchpad to host the project and Open Build Service to build your project, but over time you will need more control and do this yourself.

You need a good team and active community

- A designated active leader (Linux) or active leaders (MariaDB / PostgreSQL) that have the respect from the community.
- A group of good open Open Source Citizens maintaining the community.
- People with good "old time" coding style standards that actively teach others and participate with the community.
- Active and passionate user and developer community around your product.
- Developers that is using the product daily in production.
- Developers that need to extend the product for their own needs.

Generally the quality of the product is defined by the technical leaders

Transparency is critical for long time success

- An open development model (all email list, discussions architecture, decisions should be open).
- Clear guidelines for how things are done and will be done.
- Clear license and business model.
- Extensive documentation.
- Be clear about your roadmap and release schedule.
- Good open review process of patches that enforces quality.
- Be transparent with your plans and let users influence them.
- Be open about your bugs, and when they will be addressed.

Keep your promises!

Communicate with your community

- Attend conferences and talk about your product.
- Listen to what your users want and either do it or help them do it themselves.
- Make it easy for people to communicate with you and find information about your product!
- Ensure that most questions on your forums and email lists are answered.
- Understanding that people who write the code 'owns' the code
- Don't expect the community to produce any code.

Recognize that the path for turning a user to a customer can be long, and don't let short term monetary interest distract your long term winning strategy

Be good open source citizens

- Be open about your use of external code - Share the fame!
- Provide quick feedback to providers (bug reports and fixes, code enhancements, ideas)
- Help others in forums (around your code, and around related code you know)
- Keep up a high ethical standard on the email lists/forums.
- Don't talk bad about competing projects.
 - > Fact based comparison is ok.
- Being a good Open Source Citizen will build trust in you and your product, and will pay off over time.

Building a supportive Community later in the game is extremely hard - You should start day 1

You need to get the product out there and used!

- Release early - release often, binaries + source.
- Aim that each release, including alpha, should be bug free enough to be usable in production by those that try it.
 - > People will first use your product and only start extending when they believe in it and it's easier to extend it for their own needs then move to something else.
- Work with the distributions and cloud providers to get the product in there.
- Use a good open toolset for development (Sourceforge, Launchpad or own) to make it easy for others to participate.

Doing business with Open Source

Open Source is a philosophy and a development model.

Open Source does **not** gurantee that you will get enough money to develop and/or support your product.

Different kind of Open Source products

- Products developed by the community.
 - > Typically larger projects that are developed by many companies to solve their business critical needs or to embed in their products.
 - Linux, PHP, Apache, MariaDB
- Products that are a tools for the company and released as open source to get more community development.
- A small team or company that is developing and driving an Open Source product as their main offering.
 - > These have often a hard time to compete with closed source software (not enough money for full time developement).
 - MySQL, JBoss, Wordpress, most open source software

Questions to consider when creating a company

- Do you plan for a virtual company (“no offices”)?
- Are you creating a company of equals? Should the company be owned by employees ([Hacking business model](#))?
- Do you want to concentrate on services or development ?
 - > If development, what license to choose?
- Do you plan to have a big community or work with a few big companies ?
- Do you plan to take in investors ?
 - > If yes, then you need an 'exit' plan.

What business model to choose?

- Service company
 - > Man powered services (support, training, consulting)
 - > Valuation 2 x revenue
- Software company
 - > Licensing, Software as a service (SAAS), Subscription
 - > Valuation $10 \times \text{revenue} + X * \text{number of users}$

What is the final goal with the company

- Sold on the market
 - > Unpredictable future for product/employees
 - > Quick, often high, profit for owners
- Go public on stock market
 - > Original owners can still be part of steering the company
 - > Somewhat unsure profit (as owners can typically only sell after 6 months)
- Owned by founders, employees (and investors)
 - > Stable predictable future
 - > Owners gets dividends, employees gets bonus
- Create an open source foundation

Why go Open Source?

- Spread the product more quickly (**more users**)
- Get some part of the development done elsewhere (**lower cost**)
- Get things more tested and more bug reports (**higher quality**)
- Possible to get development done in 'not business critical' directions (**more useful product**)
- Easier to find good developers, partners and customers

The above means more market recognition, feedback, leads, business, partners and sell opportunities and a strong trademark.

In general open source projects gets **more feedback** and **better bug reports** than closed source project.

Reasons for users to trust a open source vendor

- Open source vendors are more trustworthy as they depend on trust to survive
- **No vendor lock in.** Your investment in using the product is safe even if:
 - > Vendor goes out of business
 - > Vendor would surprisingly change business terms radically
 - > Vendor would stop supporting the version of the product you are using

If this would happen to a popular product, someone would go away with the code and start maintain it themselves

- Little risk for **hidden trap doors** as one can examine the product code

Benefits for developers using open source vendor

- Easy to get access, look at and use the code
- Freedom to examine and change any part of the code to satisfy your business reasons, fix bugs or port to other systems.
- Freedom to find anyone to do the above
- Freedom to use (read, build and change) the code and redistribute the code in an open source environment.

Benefits for big business / countries using open source

- You can develop your own infrastructure to tune the software for your own means (language, unique requirements)
 - > Facebook, Google
 - > Brazil, Iceland
- You get internal knowhow you can utilize for business
- Not depending on an outside vendor (for minor things).
- No license costs; Low to very low cost of ownership!
- It's in your interest to collaborate with the original community for long term sustainability
 - > A full fork is expensive in the long run!

When go Open Source?

IF you can create a sustainable business model around Open Source, proprietary vendors will have hard to compete with you

It's nowadays increasingly harder to sell closed source solutions in a lot of business sectors.

Nor surprisingly, it's hard to choose the right license for your software as it significantly impacts your strategy, i.e.

- How you can co-operate with your user community
- How you can build your business

How to choose an OS license?

- Key Questions:
- What is your business idea around the open source product?
 - > Services, subscriptions and/or licensing ?
- What rights do you keep to your code when used, modified and potentially redistributed?
- What kind of community do you want around the product?

The main open source licenses (Very simple view)

- **Public domain**
 - > Gives user freedom to do anything, including changing your copyright and claim they wrote it.
- **BSD/Apache**
 - > Gives user freedom to full use, but needs to keep copyright in the source code.
- **LGPL**
 - > Gives user freedom to use freely, but if they distribute it changed, they need to publish the changes under LGP
- **GPL**
 - > Gives user freedom to use it freely, but if they distribute it, they need to publish changes and their code under GPL
- **AGPL** (Optional addition to GPL V3)
 - > Free usage, but users needs to publish the code and the code connected to it even if not distributed (like web server).

Business models to use with open source

- Open-Core Model - have an open-source core and sell closed-source features on top of it (e.g. SugarCRM)
- Dual Licensing Model- one product/project that gets licensed with a viral, GPL-style license and a commercial closed-source license (e.g. MySQL)
 - > Another option is “**Business source**” or delayed Open Source
- Services Models - where you get to download a productized version of an open-source project and pay a fee for the support you get on it or the features you sponsor. You can normally also pay for training, features etc
- Subscriptions (Usually a combination of support, extended product lifetime and guaranteed updates)
- Creating a non profit foundation to fund the development (FSF, MariaDB foundation)

Open Core

- Probably most popular way nowadays for business trying to do open source first time.
- What Oracle is doing with MySQL and EnterpriseDB with PostgreSQL
- Is not an open source business model, because it uses closed components and most of the benefits open source developers expect from the product is gone:
 - > You can't change, modify, port or redistribute the code
 - > You are locked to one vendor
- You may be able to create a small developer community around the product but mainly by people that doesn't need the closed source extensions.
- For community developers, the “worst” possible offer is open core or closed source that used under subscription and stops working when subscription runs out

Dual licensing

- Used first by Ghostscript. MySQL was the second product to use it.
- Can only be used when you have full rights to all the code.
- Give out the same code under two licenses, for example GPL and normal commercial closed source.
- Companies that can't use the GPL (because they don't want to give out their code) can buy the closed source version.
- Only works well for infrastructure, easily embeddable products, like libraries or databases.

Business source

- Not an open source license, but gives the community similar advantages.
- Source code is available, anyone has the right to copy, modify & distribute but can't use it commercially under some conditions you define.
- After X years the code automatically converts to some open source/free license. The date has to be explicitly set in all source code files to avoid misunderstandings
- Better than Open Core as this removes the “one vendor” problem and the code will eventually be free.

Business source

- Published in <http://timreview.ca/article/691>
- Overview at <http://monty-says.blogspot.com/>
- Discussed on Slashdot
<http://developers.slashdot.org/story/13/06/26/1552215/monty-suggests-a-business-friendly-license-that-trends-open>
- Unfortunately the second link is blocked in China, so you have to access it by a VPN.

The importance of selling licensing

- MySQL would never have been possible without licenses.
- Very hard to get companies to pay for support or development
 - > Companies expect to use open source for free
 - > Exception is companies you work with to develop the open source project
- Subscriptions are good, but also hard to sell
- Licensing is “free money” for the project
 - > You need one entity that holds the copyright to the whole project or use SAAS (Software as a service)
 - > You need to be able to dual license your project
 - > Your project is an infrastructure project that is usually embedded into others.

Business differences between MySQL and MariaDB

- MySQL Ab made money on services (support, training, paid for development) and **dual licensing**. Designed to be able to take investors.
- MySQL was mainly developed by one company.
- MariaDB has companies like SkySQL that does services around it. MariaDB foundations gets donations to drive MariaDB development with the community.
- A lot of development on MariaDB is done by SkySQL, but there is also a lot of community development.



The end