



**How to create a
successful open
source project.
St. Andrews 2012**

**Michael Widenius
MySQL & MariaDB founder
monty@askmonty.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.

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 from others while doing it.
- You participate in it 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 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 gitbub or launchpad to host the project and Open Build Service to build your project, but over time you will need more controll 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 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.



The end