

Price

Software should not be expensive!

There are 2 extreme approaches for distributing software:

1. The commercial approach:

The software is first created. Then it is sold by a commercial entity. The source code is secret and not distributed. It can only be modified by its creator or his partner(s). Thus, the creator has full control over the product's evolution and price. Typical examples of this approach are commercial companies like Microsoft.

2. The open-source approach:

The source code is not secret. It is freely available from the beginning on and everyone can participate in the development. Everyone can use, modify and redistribute the software, including the source code, under the terms of license. This approach has become very popular since the nineties. Examples of open source products are Linux, Apache web server, StarOffice, etc.

Remark: Several slightly different interpretations exist for the term 'open source'. Open source does not necessarily mean 'free of charge', although this is the case for most popular open source products. An official definition for 'open source' can be found at <http://opensource.org/docs/definition.php>

Open source development has some unbeatable advantages:

- **Huge developer community.**

Thousands of brilliant brains all over the world contribute to continuous improvements by communicating good ideas, writing pieces of code, testing new versions, correcting errors, writing documentation, and providing online resources and help.

- **Rapid progress.**

Early and continuous feedback from users tremendously contributes to early error detection, which is, as we saw throughout this document, one of the most important conditions for creating reliable software. Open source is an ideal way to get quickly informed about software bugs and missing functionalities. Thousands of people testing the software under any special and unusual conditions and configurations is much more efficient than a small group of programmers testing only the cases they have thought of. Moreover, it is much less time-consuming and much cheaper to correct errors in an early state, rather than after deployment and delivery.

- **Low price.**

Because most developers are not rewarded with money for their work, open-source software is typically much less expensive than commercial software, even free of charge in numerous cases.

A very interesting and important question about open source development is: How is it possible that people work on a project when they are not rewarded with money?

The experience shows that programming is an irresistible fascination, able to reach mania, particularly for the best programmers if they can realize their own ideas and if they are conscious of working for a beautiful, high quality, professional product that will be used by an important number of people. Most programmers experience that good feeling of having found the right solution after long moments of brain-storming: well designed, nice and smart code. This particularly satisfying feeling for a beautiful creation exists for most artists. I always remember a musician (I can't remember his name) who, when asked the question "What were the most important moments in your life?", answered: "Sometimes I'm searching for the right note. It can take hours or days to find it. But then, at the moment I've found it, I'm the happiest man in the world!". Another example is my grandfather. He was a painting artist. Painting was his life, up to the point of his profession being an obsession. I will never forget the moment when I was a 6 years old boy seeing him very, very ill, sitting on a chair and painfully trying to move his trembling hand, holding a brush, to an unfinished painting. Before dying, he told my grandmother his last wish: being able to paint for 2 more years. He never bothered about money. Without his "financial manager", who happened to be my grandmother, he probably would have been a poor man (seen from the material point of view).

Obviously, there are other rewards besides money: fun, fame, appreciation, pride, prestige, social contribution, passion, obsession, etc. They generally excel money, under the condition that survival is guaranteed.

However, the open source approach also suffers from some drawbacks. Why?

It is a fact that we can never force someone to do something. There is always a motivation behind someone's doing,

e.g. money, any kind of good feelings and fear. The problem with open source development is that the motivations for working without being paid are limited in time. Developing new software and realizing personal ideas is very pleasant at the beginning. But this great joy risks disappearing after a while. Once a first version is presented to the users, they ask for improvements: additional functionality, error corrections, better documentation, easy deployment, and so on. These user requirements typically end up in a big to do list with tasks that are much less pleasant and sexy than inventing something new. And this is the point where the open source way may fail. The initial sources of motivation are gone. Discipline, hard work and professional consciousness are now required. Hence, a replacement for the initial motivation must appear. In most cases, this can only be money. Moreover, no one can work for a long time without being paid. If a developer spends most of his time on a project, she or he should be rewarded with money. Besides, big projects depend on a good organization and coordination. People occupied with such tasks would probably be hard to find when there is no monetary incentive. There are also expenses for material and other costs. Hence, money is needed if we want to avoid that the project collapses after the initial bliss. Open source is never a guarantee for success. There are numerous examples of pitiful ending open-source projects. Commercial projects don't suffer from this inconvenience. Monetary incentive is present from the beginning on and therefore the above discussed risk doesn't exist. As a result, the user's confidence of getting a high quality product with continuous improvements and professional support may be much higher. On the other side, open source may considerably reduce the user's fear of being too much dependant on a supplier, because everyone can get the source code and change it, without any limitations in time.

As both approaches have advantages and inconveniences, the question arises: Isn't there an ideal way, the golden middle that supports the best of both extremes and eliminates their inconveniences? Regarding Obix, pondering about an optimal approach didn't yet end up in the elaboration of final distribution rules. However, the following principles should be applied:

- High quality and best technology for the users must predominate. Any counteractive political or commercial considerations must be suppressed as far as possible.
- Source code must be available and everyone should have the right to adapt it to his personal requirements.
- Everyone can actively participate in the development and improvement of the product. A well organized developer community should exist.
- Forking must not appear. At any time there should be only one official version.
- The software should be free of charge for non-commercial uses. But anyone making money with a product should also participate in its development costs.
- Fast and professional support must be available (see next page).

Any comments, critics or other ideas about the above thoughts are very welcome.