

Open Source - Free Market Capitalism of Technology

✘ Before I get started, I want to say that I am a huge supporter of open source technology, open standards and I firmly believe that using open source to innovate is one of the most advantageous ways to advance a software or infrastructure product.

As I bring more and more people into great open source technology projects such as Puppet and OpenStack, there are a few challenges that can become a barrier to certain individuals, and their organizations, in adopting the tools. Most often, the issue comes with support and the development roadmap. What comes to the surface quickly in these cases is that the openness does not automatically imply support and adaptability.

With proprietary, closed source products (e.g. vSphere, vCenter Orchestrator, SCCM, Hyper-V) we have a fairly fixed set of support and the drivers are listed for known hardware and software compatibility. Additional support can be requested as a feature request, but generally the packaged product is a WYSIWYG (What You See is What You Get) technology and we work within its limitations.

The hot term lately is the “one throat to choke”. It’s the idea of a vendor model where you have a single place to contact for purchase and support for major portions of your infrastructure. This is usually only available in closed source, or what are called “walled garden” types of products and services.

Open Innovation

Using open source technologies, you have the ability to use the known drivers and interfaces, plus there is the bonus of being able to develop the product and advance the technology as you need to. But the challenge is that word “develop”.

For many people and organizations, developing drivers, shims and other enhancements into a product may not be in their capability. These skills are over and above what many can effectively do.



The community is active, but there is no guarantee that the particular feature or driver you are in need of is in active development. A road paved with hope is a dirt road, so there is no way to force the integration of the product feature that you are after, and there may even be risk of regression of current support and compatibility as the core product changes.

Free Market isn't the Panacea

In the economic world, there is a strong support for the use of free market capitalism to grow the economy. This is absolutely the best method to significantly advance the economy and flourish throughout the world...in theory. In practice however, the idea that if we reduce taxes on high earners to allow them to put money back into the system for all to use is flawed by the human condition.

The proven history shows that the money doesn't flow down in the way that the theory has dictated it should, and of course, there are all sorts of people to blame for this from all sides.

On the other side of it, the goal of free markets is that they can also freely fail. It is entirely possible for the economy to tumble for a number of reasons, and the conundrum is built where the owners of the wealth want to be protected from failure.

How does this compare to open source development though?

It's your fault the book isn't finished

Imagine a book that was given to you with 200 pages in it, but only 140 of them are completed and the story has missing components that you would have thought to be required. You approach the author on the issue and in response, they hand you a pen and say that you have to finish the book, and it's your fault that it isn't good right now because you haven't put the time in to finish it.

Even better, you write the finale to the book, and then a sea of complaints on your choice of writing come in to your inbox. The open source methodology has now made you a part of the system; but just like free market capitalism, it may not always work in your favor.

Quite often we find that there is a "works in my environment, so it must be localized to your configuration" statement, and you are left searching for answers to make the product fit correctly for your needs.

So in choosing a product based on its potential to be grown, adapted and advanced, you ultimately are still left with the basic abilities of the core and unless you take on the task, or find someone who can do it for you. As a result, the product will not innovate in the way that we have been led to believe that it can go.

Fragmentation Nightmares

The Android OS is a prime example of one of the key challenges in an open source environment. Each telco has branched off a piece of the code and is deviating from the core in order to support their very specific needs and lock you into the version in order to reduce their support impact.

The very thing that we loved about open source has been broken in this case. The openness has been relinquished for vendor lock-in.

The Forgotten Ones

Another challenge is that you are working with a plug-in, module or drive that had community development, but you look at the last commit date in GitHub: December 2011. Much has happened since then, and with some core system update this custom update no longer works, but the project owner has abandoned it.

What are we getting to with this?

To summarize, I wanted to highlight the highs and lows of open source products, and what the impact may be in your environment. I am pretty sure I'll take a beating from a few with some of the more general statements made here, but it is important that you, as an architect, are aware of risk factors.

The FUD flies both ways of course, and the walled garden is no panacea either. So when you make your decision, just flavor to taste and be aware.