

The Symbiotic Relationship Between Software and Globalization

Gary Kim

October 6, 2020

Word Count: 2817

Contents

1	Abstract	3
2	Introduction	4
3	Open Source Software	6
4	Single Globalized Software Market	8
5	Software in Developing Markets	10
6	Tragedy of the commons	12
7	Conclusion	14

Abstract

Globalization has had an affect on nearly every industry. Globalization allows economies to engage in trade with each other, so economies can more optimally allocate its own resources, resulting in a net increase in productivity, allowing economies to produce far beyond their normal production possibilities curve and therefore improve the GDP and living conditions of the citizens.

In this paper, we look at how those effects have manifested itself in the software industry and how the software industry helped to enable globalization. The goal of this paper is to ascertain what kind of effects the software industry and the process of globalization has had on each other and other possible side effects and whether the relationship has been beneficial for both, one, or neither software and globalization.

We find that globalization and the software industry have had a mutually beneficial relationship as they evolved in the last few decades. Globalization has helped the software industry grow and become more successful through the single globalized software market and open source software. The software industry has helped to accelerate globalization in developing countries helping to better connect the world's economies.

Introduction

The software industry has a unique relationship with globalization. Globalization, as we see it today, would not have been possible without software. In addition, while the software market was mainly an American venture at its inception, globalization would completely change the direction in which the industry would go. Several articles and books have looked into how the software market and globalization have affected each other.

Arora and Gambardella of Carnegie Mellon University and Sant' Anna School of Advanced Studies looks into how developing countries have adopted software to great benefit as the industry has developed to become a big employer in their economies, largely by exporting to other countries, in their article *The Globalization of the Software Industry: Perspectives and Opportunities for Developed and Developing Countries*. Arora and Gambardella explores the spectacular growth of the software industry in countries such as Brazil, China, India, Ireland, and Israel. It finds that software is now a large part of the economies of many of these nations, helping raise the GDP of each nation by billions. It also looks at the effects of the growth on the software industry of the United States. The researchers found that the rise of these new software-producing regions will be of benefit to the United States and the US does not need to fear losing its technological leadership due to these new regions anytime soon.

Reilly, Smith, and Benkler, in their book titled *Open Development: Networked Innovations in International Development* touch on how firms are now capable of globalizing and leveraging economies of scale to an extent never before imagined. Reilly, Smith, and Benkler also explores how globalization of developing countries is significantly accelerated thanks to connectivity that software helps to provide.

Campbell-Kelly Martin of the University of Warwick and Naiel D. Garcia-Swartz explores how the hardware and software industry globalized in their book *From Mainframes to Smartphones: A History of the International Computer Industry*. The authors found that America still plays a dominant role in the software industry but other countries such as Israel and Ireland have

also become influential players in the industry (MARTIN and GARCIA-SWARTZ).

In this paper, we look to add on to the already existing content with an analysis into the specific ways in which the software industry and the process of globalization has helped each other from an economic point of view.

Open Source Software

Open source software development is the practice of making "a body of original material ... publically available for others to use, under certain conditions" usually involving making any enhancements available to other users of the software (Lerner and Tirole). There are many examples of large software projects developed with this model, the best known of which is the Linux kernel. The model of developing software in this way "began the late 1970s when Richard Stallman of the Massachusetts Institute of Technology (MIT) launched both the GNU project and the Free Software Foundation" (Reilly, Smith, and Benkler). This movement was then significantly helped by networking and globalization. Collaboration for the open source development model "has become infinitely easier thanks to the Internet"(Klemens).

The Linux kernel is a good example of an open source project that has really benefited from globalization. The list of current top companies that contributed more than 1% of the changes between v4.8 and v4.13 to the Linux kernel include Intel, Red Hat, Samsung, Huawei, Canonical. In just this list, Intel and Red Hat are American companies, Samsung is a Korean company, Huawei is a Chinese company, and Canonical is a UK company This is according to the "*2017 State of Linux Kernel Development*" which is a report the Linux Foundation occasionally releases as a look into how development of the kernel is progressing. Globalization and the network made by it has allowed for collaboration to a degree that would not have been otherwise possible.

Software developed in this format has provided a lot of value to everyone. The existence of open source software means that firms already have a lot of capital to begin with when building software products which helps to improve the production possibilities curve of software firms in

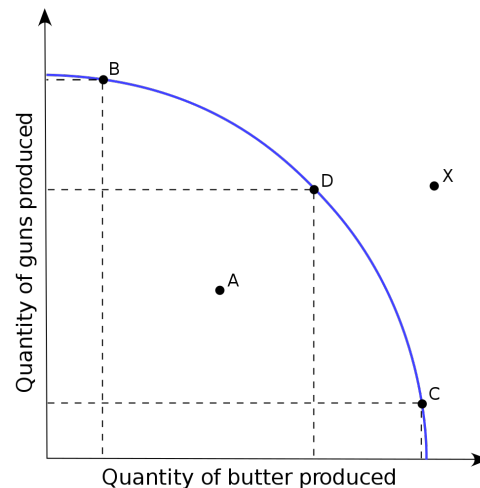


Figure 1: Production Possibilities Curve, Wikimedia User Everlong, CC-BY-SA-2.5

all economies. Figure 1 shows a production possibilities curve. Generally, it is not possible to produce outside the curve without trade. Open source software has a similar effect to trade, allowing firms to produce at points such as point x. The magnitude of the effect that open source software has becomes clear when looking at the penetration of open source software in establishments. "only 18 percent of establishments in Sweden", "31 percent...in the United Kingdom", and "44 percent...in Germany" "use some sort of open-source software". Many of these firms benefit so much from open source software that "thirty-six percent of the companies surveyed" say that their "software developers are free to work on Open Source projects within their time at work" (Klemens).

Open source software has also been shown to have the effect of improving the value of commercial software through providing capital and increasing competition. "Consumers were better-off with OSS competition when it increased the intensity of price competition" (Atal and Shankar). Beyond the direct benefit provided to individuals in the form of free to use and modify tools, open source software also acts as competition for software vendors. In addition, many software firms are built on open source so firms also benefit heavily from open source software that allows them to produce value far greater than would have been possible if everything was built from scratch. This combined with the price competition give by open source projects create interesting cases of firms that build on already existing open source software and earn money from selling their improvements. Red Hat is a great example of this, being one of the largest open source companies whose Red Hat Enterprise Linux product is built on top of the open source Fedora project. Red Hat benefits from the work and capital of the Fedora project while consumers benefit from the price competition of the Fedora project according to their "*What is the relationship between Fedora and Red Hat Enterprise Linux?*" document.

Globalization has helped to start an entire new category of software that has helped the software industry achieve a level of value and productivity that would not have been possible otherwise.

Single Globalized Software Market

Globalization has had the effect of making a single market for many goods and services provided by firms but the effect of the single market has been strongest on the software industry due to software's blindness to the physical location of the consumer and producer of the product or service.

Typically with trade and globalization, overall surplus is increased in economies thanks to comparative advantage and more competitive firms, as is represented by Figure 2. In Figure 2, globalization and trade means that firms in the two economies involved are now essentially in a single market and in competition with each other. For the market in Figure 2, Brazil, on the left side, has a comparative advantage in the production of sugar so the consumers in the United States get the benefit of lower pricing by firms in Brazil while firms in Brazil get the benefit of higher pricing for consumers in the United States, resulting in an overall increase in surplus represented in the graph by the blue shaded area.

Something that Figure 2 does not show is the cost of doing international business. Due to costs such as shipping and taxes, the demand supply graph tends to look more like 3 where the surplus ends up being smaller and local firms or consumers seem better, even if they are not from a pure value perspective, due to the extra cost of doing international trade. While software products or services traded overseas are also subject to taxes, costs such as shipping are virtually non-existent. Because of this, globalization has allowed consumers and firms in the software industry to benefit more from the increased surplus of international trade than many other industries.

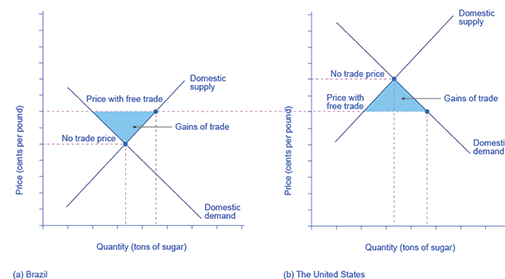


Figure 2: Demand and Supply with Trade, lumenlearning.com

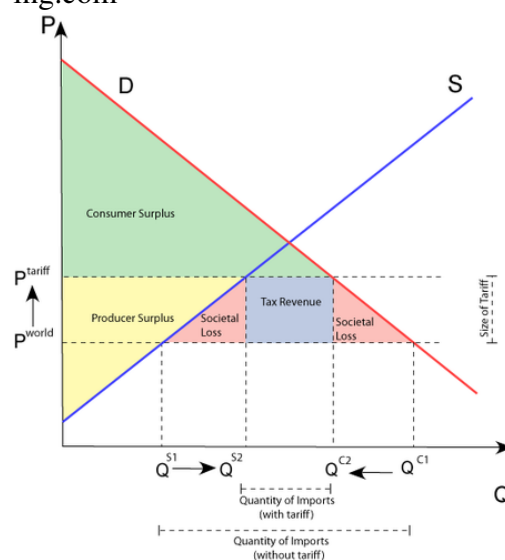


Figure 3: Demand and Supply with Trade and Tax, Wikimedia User Austin512

The single market has helped to drastically improve the effects of some practices in the software industry. Ben Clemens in their paper *Math You Can't Use: Patents, Copyright, and Software* describes how “if a company is not hoping to make big profits from a piece of software, its best bet is to go to the other extreme and open the code base entirely, allowing for free and open collaboration” (Klemens). This would likely have been true with or without globalization but the single market has helped to multiply the positive effects of this as well as how common the practice is. With such a large market, even small software projects that companies start for their own uses can be made into a product used worldwide.

Globalization has the effect of combining markets to improve overall surplus in economies but this effect has been especially significant on the software industry.

Software in Developing Markets

Globalization has had varying effects on developing countries. It is often debated among economists whether or not globalization has had a positive or detrimental effect on countries. It has been shown that for some countries, the introduction of international trade has stunted the growth of the economy while others have shot up becoming some of the fastest growing economies of the last few decades. The introduction of software has made its own unique spin on these effects.

The software industry has become an integral part of the economy in several nations (Arora and Gambardella). Ashish

Arora and Alfonso Gambardella discussed how “During the 1990s India, Ireland, and Israel emerged as significant software exporters. In the same period, Brazil and China also developed an extensive software sector relying largely on the domestic market and are now attempting to move to exports.” The importance of the software industry to their economies can be seen in how large they are in comparison to the whole economy. For Brazil, China, Israel, and India, software sales make up between 1-4% of the country’s GDP. These countries also have a very high export of software being as high as 70% to 85% of sales for India, Ireland, and Israel.

One of the reasons for software becoming such a large part of these economies seems to be from these economies having an abundance of technically skilled workers relative to the demand from manufacturing and related services. “One regularity...is that they exhibited an excess supply of human capital in the 1980s and early 1990s, and specifically an excess supply of engineering and technology graduates” (Arora and Gambardella). Software firms, thanks to some of the reasons covered before such as open source software, are particularly easy to set up so in countries that don’t have “plentiful job opportunities in industry, in well-established firms with good opportunities”, “entrepreneurs” tend to set up more software firms.

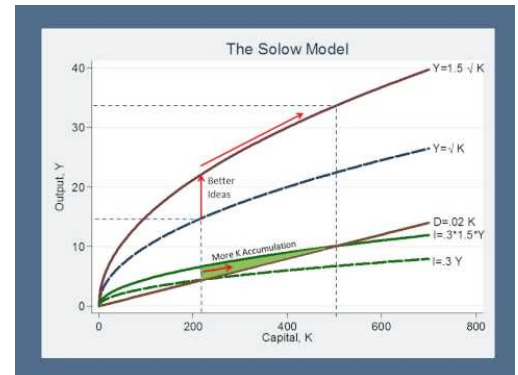


Figure 4: Solow-Swan Model, mru.org

Another way to look at it is in terms of the Solow model. The Solow model describes long-run economic growth of an economy through how the economy gains capital. It says that similar economies will head towards a similar equilibrium. Essentially, the Solow model says that output of economies will increase by investment in capital and technological advancement. In Figure 4, this is shown as the increasing value of output (Y) as Capital (K) increases. The model also in part implies that due to diminishing returns, developing economies will catch up to developed economies rather quickly. The software market is very helpful for this catch up. Software is relatively low capital and the existence of open source software means that a lot of the capital is publically available. This means that developing countries can benefit a lot from adopting software as an integral part of its economy to help grow more quickly than is possible with other industries.

The software industry has become an integral part of some developing economies and helped to quickly increase the GDP and living conditions of people living in those economies.

Tragedy of the commons

While broadly, both the software industry and continued globalization of the economies of this world have benefited each other, Atal and Shankar does raise an interesting counterargument in regards to the benefits of globalization or in this case more specifically open source software for the software industry. They argue that OSS results in the “classic public good under-provision problem” due to the fact that in open source software, “quality of software does not depend on [number of consumers] since each user only cares about her own value and not the value that her contribution generates to the other users in the network”. The tragedy of the commons describes a situation in which a shared resource will be quickly depleted as everyone has an incentive to exploit the resource without taking the necessary steps to safeguard the continued use of the resource. A common modern example being fishing in which every fisherman has an incentive to fish as much as they can with has led to an overfishing issue in which we are now causing the endangerment and extinction of many fish species that, if the tragedy of the commons could have been avoided, we could still be exploiting for benefit. Common good in the form of products of digital technologies is referred to as “Knowledge Commons” (MULGAN).

This concern is very valid and we have seen, at times, this prediction playing out exactly as Atal claims it would. This eventually can be and is avoided by many projects, though. Copyleft is often used in an attempt to avoid the tragedy of the commons. As Mikko Mustonen of the University of Helsinki, Department of Economics, covers in his paper titled *Copyleft-the economics of Linux and other open source software*, “copyleft licensing creates an alternative incentive structure reminiscent of scientific research” in that it requires “the licensee [of software] also grant similar rights over the modifications he had made” (Mustonen). The idea is that copyleft acknowledges the tragedy of the commons issue and addresses it by legally requiring that improvements become public for everyone. The unique thing with

Mustonen also analyzes why open source software can continue to exist in our economy alongside traditionally licensed software products. He finds that economically, “when the cost is

sufficiently low, some consumers choose to use the copyleft program and the monopolist". There are situations in which open source software is actually more beneficial to contribute to for firms rather than creating their own product, also helping to reduce the potential of the tragedy of the commons.

Conclusion

Globalization has helped accelerate development and increase the value of software and the software industry has also helped to increase globalization and improve various economies. The interconnected world that globalization helped to create has allowed open source software to flourish far more than it would have been possible without, helping provide benefit to many individuals and providing pre-made capital for firms. The single global software market created by globalization helped to increase competition and value to a degree that is more significant than that of most other industries, helping increase consumer surplus. Software, being a rather easy to adopt industry with a lot of capital available publically in the form of open source software, has helped push a few developing economies into productive economies in our globalized world.

The ideas that have worked in the software industry could be adapted to work in other industries to help to take better advantage of the benefits provided by international trade and globalization, helping to stretch the production possibilities curve of economies much farther than it would have been with purely market driven allocation of resources. The extensive use of collaboration and reducing redevelopment of capital and technology in a way that benefits everyone while still keeping the return on investment can be of help in some other industries. This could be of help to both increase the world's productivity and to bring more developing countries into the globalized world.

Works Cited

- “2017 State of Linux Kernel Development”. The Linux Foundation, 2017. Web. 18 Nov. 2019.
- Arora, Ashish and Alfonso Gambardella. “The Globalization of the Software Industry: Perspectives and Opportunities for Developed and Developing Countries”. *Innovation Policy and the Economy* 5 (2005): 1–32. Web.
- Atal, Vidya and Kameshwari Shankar. “Open source software: competition with a public good”. *Atlantic Economic Journal* 42.3 (Sept. 2014): 333+. Web.
- Klemens, Ben. “The Decentralized Software Market”. *Math You Can’t Use: Patents, Copyright, and Software*. Brookings Institution Press, 2006. 92–107. Web.
- Lerner, Josh and Jean Tirole. “The Economics of Technology Sharing: Open Source and Beyond”. *The Journal of Economic Perspectives* 19.2 (2005): 99–120. Web.
- MARTIN, CAMPBELL-KELLY and DANIEL D. GARCIA-SWARTZ. *From Mainframes to Smartphones: A History of the International Computer Industry*. Harvard University Press, 2015. Web.
- MULGAN, GEOFF. “The Rise of Knowledge Commons: It’s for Everyone”. *Big Mind: How Collective Intelligence Can Change Our World*. Princeton University Press, 2018. 200–214. Web.
- Mustonen, Mikko. “Copyleft—the economics of Linux and other open source software”. *Information Economics and Policy* 15 (2002): 99–121. Print.
- Reilly, Katherine M. A., Matthew L. Smith, and Yochai Benkler. “The Emergence of Open Development in a Network Society”. *Open Development: Networked Innovations in International Development*. Mit Press, 2013. 15–50. Web.
- “What is the relationship between Fedora and Red Hat Enterprise Linux?” Red Hat, Inc, web. 18 Nov. 2019.