In March 2013, a struggling start-up announced that they were open-sourcing their core-technology, which enables developers to move software easily between different machines by enclosing it in “containers”. What started then arguably has no precedence as Docker has become a phenomenon in the open-source community. With over a billion downloads, nearly two thousand contributors, and $150M funding, Docker has become a main player in the software landscape. But none of this would have happened if they had not bravely open-sourced their technology.

We hear this term "open source" almost ubiquitously nowadays, but what exactly does it mean? More to the point here, what is the “open source community”? There are many attempts to define “open source” and they certainly pay reading. However, it is also instructive to consider a negative approach by asking what is “closed source” software. When developers take their code and compile it, they render it in binary form, ones and zeros; a form that computers can actually work with. However, it would take a most unusual mind to be able to make any sense of such a data stream. Given the speed with which processors work, this is in fact an impossible task. So, what do we do when our apps do not function as expected? We report a bug and a software maintenance guy is called in. He or she goes back to the un-compiled code and fixes it. In other words, it’s a closed process.

On the other hand, "open source" refers to code that can be modified and shared because it is available to everyone to modify or enhance as desired. Open source software licenses promote collaboration and sharing because they allow other people to make modifications to source code and incorporate those changes into their own projects.

But the benefits of open source are even greater yet. Licensing has been used by software providers to create massive corporate behemoths which dictate terms of use (End User License Agreements, EULAs) which we either accept or decline. And we pay for. Open source software usually also requires users to agree to terms of use, but without charging a licensing fee for it. In other words, computer programmers outside the original development environment can access, view, and modify open source software whenever they like—as long as they let others do the same when they share their work. In fact, they could be in breach of the terms of the user license if they do not comply.

Just as the translation community has found that there is often more than one way to say the same thing, there are different ways of writing computer code to implement an algorithm. Aesthetics as well as practicality kick in and coders find there can be better ways of solving a problem. Also it became apparent that it was inefficient for certain routines to be coded from scratch in different programs. Why reinvent the wheel? Reuse of code therefore became a hot issue. But who owns the original code? To cut to the chase, the implications of facing innumerable disputes over ownership resulted in the advent of community ownership. These days, you’d be hard-pressed to find a corporate business that has not embraced open source. Yet even now, open source can trigger skepticism. Its lower cost and the apparently free-spirited approach of providers - its very openness - can inspire disdain. But adopters citing
benefits such as better innovation, greater flexibility and reliability are winning the day. No one can seriously doubt open source software’s prevalence as it challenges proprietary exclusivity across the IT world. It is not only our day-to-day IT infrastructure that is facing change, open source has fostered further innovation in cloud computing and global networking. Other developments in internationalization tools, the analysis of data and the emerging Internet of Things indicate that the scale of open source’s adoption will only increase and provide us with greater potential to enrich our technological domain.

The language industry has its own fair share of open source software. A quick look at Sourceforge reveals 250 localization applications ranging from full multilingual translation-management applications all the way to language-specific plug-ins. The Free Software Directory under Localization has nearly 70 applications with some well known as Apertium and Okapi.

Moses is synonymous with machine translation. What is not so well-known is its origins. In 1997 Philipp Koehn, a German national, started his PhD thesis at the University of Southern California on knowledge-based machine translation. By 1999, a small core group of researchers from select universities started working towards a statistical model of machine translation. Koehn was highly involved in the new “statistical wave” and he is one of the inventors of “phrase-based” machine translation, a sub-field of statistical translation that employs sequences of words, “phrases”, as the basis of translation, expanding the previous word-based approaches. A paper entitled Statistical Phrase-Based Translation, which he co-authored in 2003 with Franz Josef Och, of Google Translate, and Daniel Marcu, has attracted wide attention in the machine translation community. Koehn created Pharaoh, a machine-translation decoder, as part of his PhD, which he tried to share openly with the community, but was only allowed to share the binary code. Although he moved on to the Massachusetts Institute of Technology for postdoctoral research, he continued improving the Pharaoh code.

In 2005 Koehn joined the University of Edinburgh as a lecturer in the School of Informatics and oversaw the thesis of the then Statistical Machine Translation PhD student Hieu Hoang. Koehn gave him the Pharaoh code to improve but they soon decided to re-write the entire code from scratch. The Moses machine-translation decoder was created as an open source project and was developed during the time that Koehn and Hoang were at the University of Edinburgh and during a summer workshop at John Hopkins University.

Since its initial development in 2005, many contributors have come on board and there are currently over 1,000 people on the mailing list with a core of 100 contributing developers. Hoang is by far the main contributor to Moses having written over 40% of the code. Now Moses is a faster decoder, with multi-thread training; it is now easier to install in Windows, Mac and Linux; and it also incorporates multi-stage testing. These are just some of the many major improvements in terms of functionality, usability and reliability. With both EU and corporate funding, Moses is the most widely used, state-of-the-art open source software for statistical machine translation. Both Hoang and Koehn are still overseeing, guiding and maintaining Moses.

Every large internet corporation has to develop their own software for automated continuous localization. Unlike all the others, Evernote has bravely open-sourced theirs.
Serge is the brain-child of Igor Afanasyev, current Director of Localization of Evernote. Evernote's initial product was available only in English and in 2008 Igor Afanasyev was tasked to adapt the software for the Russian market. Russian was chosen because Evernote had historically many Russian-speaking employees, including the founder, Stepan Pachikov. Afanasyev single-handedly built code, localized and translated the Evernote products in the Russian language. Through the following few years under the direction of new CEO Phil Libin and with a global imperative, Afanasyev headed the localization department which grew to include engineers and translators to a department of over 30 employees strong now. Right from the beginning he realized he had to automate as much of the process as possible. As is familiar to many innovators, his brief from the corporation was that he had “carte blanche” to automate as long as he did not request further funding, engineers’ time or other resources. In 2008 and within one single month, Afanasyev had the initial code for his continuous localization toolkit ready, but needed to add software to enable the translators’ actual work. He researched in the open source community and came across Pootle, an open source online translation-management tool mostly used by Mozilla and LibreOffice translation communities. Pootle would work well with Serge but its user interface needed improvement. Afanasyev contributed to Pootle’s open source code considerably and later on, Evernote's improved version of Pootle became the mainstream one. Evernote and Translate House, the organization that maintains Pootle’s development, continue to work closely together on this project. Serge is a robust, continuous localization platform as evidenced by its ability to support Evernote’s 150 million users in over 30 languages. Evernote heavily utilizes open source software and equally contributes back to the community. With such a culture, it was no surprise that Afanasyev got approval to open source Serge. After some further code clearing and writing documentation, Serge has been available to the open source community since October 2015 with an announcement made during the LocWorld conference in Santa Clara. Serge and Evernote’s entire open source localization process currently works with almost “zero maintenance”. Afanasyev explains “New strings appear for online translation every minute, and all translations are immediately integrated into internal builds of our products. With our agile development process, having a continuous localization process in place is the only way to go. This is why we wanted to share this system with the world: we know localization can be a pain point for any company, and wanted to make the entire experience much better for others in the industry.”

"Much like Docker did in 2013, one of our well-respected localization companies also bravely open-sourced a proprietary software package at a time when open source was still in its infancy for the translation world. In 2008, Welocalize acquired Transware, an LSP specializing in e-learning, and with it came a software package called Ambassador Suite, which originated from a previous acquisition of a company called Globalsight Corporation. Ambassador Suite allowed webmasters to generate multilingual sites while managing them in only one language. Within a few months of the acquisition, Welocalize CEO, Smith Yewell, realized that the commercialization of the Ambassador Suite within a finite, already-competitive market, along with the fight for a small market share, rendered the enterprise version unworkable. However, there was a lot more to be gained, not just for Welocalize, but for many more LSPs which could come onboard if the software was an open source project."
Welocalize aptly changed the name of the software to GlobalSight to honor the original innovator, replacing the underlying technology with open source components, fine-tuning the code to ensure support of the latest data formats and creating a steering committee with representatives from companies involved with the project. After successfully beta-testing the software, documentation and guides were released to the community in 2009. The GlobalSight community is an engaged group of users, translators and developers. Being a server-based extensible tool, it is relevant mainly to mid-size-to-the-larger LSPs and corporations. Making GlobalSight open source was a strategic business decision by Welocalize and Yewell. In Smith Yewell’s own words: “Innovation is one of the 4-Pillars underpinning our company, and we believe an open and extensible framework is the right approach for fostering innovation not only at Welocalize but across our industry.”

With a heavy investment of over $50 million and support from most of Welocalize’s large clients and involvement from other LSPs and developers, GlobalSight is a true translation-management powerhouse used by many global enterprises worldwide. However, the general adoption by other smaller LSPs has not reached the levels anticipated, mainly due to the complexity of the product implementation. By integrating with other open source components, such as desktop CAT tools and MT, GlobalSight is attempting to facilitate freelance involvement and adoption, subsequently moving to the next phase of its evolution.

The most popular open source CAT tool for desktops is by far OmegaT, with a loyal community of translators and small agencies. The original author of the software is Keith Godfrey, a then localization engineer married to a translator. To help his wife work with free software, he developed a prototype in C++ which he then converted to Java. The time was 2001 and this project may have never surfaced if it wasn’t for another translator, Marc Prior, who convinced Godfrey to hold on to it for a little while longer. A first release to the open source community happened in November 2002. When Godfrey retired from the project in 2003, the entire process could have stalled, if not for the involvement of Maxym Myklahchuk. A Ukrainian living in Italy, he needed to translate some projects and came across OmegaT. Under Prior’s coordination, and the help of Jean-Christophe Helary, a French national living in Japan and currently OmegaT’s Localization Manager, for user support, Myklahchuk developed various enhancements which led to the second birth of OmegaT in 2004. Prior started a website to support a small number of users and to provide some basic information, followed by a comprehensive documentation by Helary and Myklahchuk in 2005.

In 2006 Didier Briel started contributing to the code and after some changes in the developers’ circumstances he became the only person able to modify the software in 2007. He became first the Acting Release Manager, then Development Manager, and in 2014 when Prior decided to retire he became the Project Manager of OmegaT, which he is to this date. Today OmegaT is well organized and thanks to Prior, who remains OmegaT’s webmaster, has an extensive website with supporting documentation and resources. The project has a very loyal following but contributions have always been on an ad-hoc basis. When a developer or agency needs a new feature, either funding or code will be contributed. Briel explains “All the developers are also translators and OmegaT users with full time jobs. Whereas in the past we would get unexpected contributions and have a constant turnover of coders, today we have a small core of developers that we can count on.” OmegaT has no comprehensive roadmap but the OmegaT team draws
inspiration from 1100 requests for enhancements, provided by the community, of which 750 were implemented.

Anyone who doubts the impact that the open source way already has had on our lives should take a look at runaway open source success stories like the Linux operating system or Apache web server. These are prominent, but there are countless other success stories and these are being added to on a continuous basis. In fact, what has grown out of a software development initiative has blossomed into what is now known as “the open source way.” The extensibility of core ideas that lie behind open source software has resulted in what opensource.org defines as, ‘Open source projects, products, or initiatives… that embrace and celebrate open exchange, collaborative participation, rapid prototyping, transparency, meritocracy, and community development.’ If we think about how resources like Wikipedia and Creative Commons have transformed our online capabilities, we cannot fail to see how community benefits from embracing the values of open source.

It’s easy to focus on applications, but we should never forget the people, engineers, coders, entrepreneurs, activists, altruists of all description and so on, who make this thriving community the success it is. Numerous government, financial, commercial enterprises are now benefitting from the continuous improvement and the collaborative mandate that open source fosters. Now software internationalization is sharing in the success of the open source way. While we still need to use and support proprietary software developers, those who work freely for the good of all should command a monumental vote of thanks from us all. Long live the open source way!