

Guido van Rossum: The Modern Era of Python

Charles Severance, University of Michigan

Python creator Guido van Rossum talks about the growing pains—and rewards—the programming language experienced as it evolved.

The popular programming language Python was created by Guido van Rossum in 1989 and released as open source in early 1991. After the first National Institute of Standards and Technology (NIST)-hosted Python workshop in Gaithersburg, Maryland, in 1994, Python's popularity rapidly increased around the world. But when an effort scales up, things can start to get a bit complicated. In this column, we continue our discussion from last month with Guido, this time discussing Python's evolution from version 1 to version 3. Watch my video interview with Guido at www.computer.org/computingconversations.



See www.computer.org/computer-multimedia for multimedia content related to this article.

BUILDING A COMMUNITY

The 1994 NIST-sponsored workshop provided an opportunity for the nascent Python community to get together and figure out its next steps. For Guido, the workshop led to a job offer and a move to the United States:

From 1995 to 2000 I worked in the US at CNRI [Corporation for National Research Initiatives]. We worked through the growth of the Python community and infrastructure, and we created the Python website. When I started at CNRI, Python 1.3 was about to be released. While I was there, we released several versions leading up to 1.5.2, which remained the “gold standard” of Python for a very long time.

CNRI made it possible for Guido and his colleagues (known as PythonLabs) to make significant development investments to move Python forward, but some wondered if the product should move from open source to a model where some profit could be made. Guido increasingly found himself in conflict with CNRI leadership about the licensing and ownership of Python. A small open source startup company started courting Guido:



They said, "You must come work for our open source startup. You can find and hire a few Python engineers and you and your team can work on Python full time." I thought, "Wow, that sounds great," so I agreed to join the company. By the time we actually started on 15 May 2000, the dot-com bubble had actually already burst. I wasn't even aware that there was a bubble.

No one really knew if the bubble might come back or if it was gone forever, so things continued along for a while:

We spent the summer in blissful ignorance working full time on Python. We built and released Python 2. Python 2 was very important because it contained Unicode and because we needed some kind of graceful exit from CNRI.

CNRI didn't claim to own all of Python, as it had existed long before Guido started working there, but there was some disagreement as to the ownership of the code that had been added while Guido and others were at CNRI:

We engaged in negotiations with CNRI, facilitated by Eric Raymond and Eben Moglen, to make sure that Python remained open source with a license that was acceptable to CNRI's lawyers. Python still has a very awkward license due to the way I left CNRI.

As the open source licensing issues were being worked out, the startup paying Guido and his team of developers started to crumble:

We didn't even move to the West Coast—we worked from our homes. Once a week we had a

team meeting with the leadership in California from my living room. Within five months it was over—they just stopped paying us. But in those five months we had the time of our lives working together on the open source Python

2 release. We also released Python 1.6, which was nearly identical to Python 2 but with CNRI's license.

They never got their last paychecks from the failed startup:

Suddenly we were out on the street. They had bought each of us a good computer, so that was our last payment. We were informally told, "Nobody will take [the computers] away from you. They are not going to come after you."

PYTHON GETS A SECOND CHANCE

By the end of 2001, there was a great deal of commercial effort happening in the Python community, and several companies were interested in helping Python continue to move forward. Guido chose to work for Zope, a successful Virginia-based company that was heavily involved in the Python community. At this point, Guido was a little wiser from his experience with the failed startup, so he made sure to discuss the details regarding how he would work for Zope and continue to lead the Python community:

Zope said, "We have no design on the ownership of Python. We want you on our team. You

are really good programmers and the top contributors in the Python community."

Because Zope was based on Python, they benefitted from their investment in Guido's team. During the

Python is well positioned to become the premier language of data analysis and the language of choice for new programmers.

Zope era, Python continued to mature and gain adoption:

I got a job offer to work for a small startup in California. This time I did my homework a little better and it worked out well. Python just kept growing and the community kept self-organizing. What started out as a workshop with about 25 people in 1994 had become the international Python conference with 300 or 400 people. I always encourage the community to self-organize rather than look to me for every decision.

As Python grew and was being used in many new ways, people started to think about improving it. Sometimes it was easy to add an element to Python, but other times a suggested improvement could break the existing code. These inconsistencies, omissions, and errors are known as Python "warts."

THE BIRTH OF PYTHON 3

Around 2005, the accumulated list of warts led to the realization that Python needed a major revision. This would be a huge effort because the revision wouldn't be backward compatible and would break the ever-growing set of Python 2 applications. The idea was that non-backward compatible

changes should be done only once and at the same time:

Early in the first decade of the new century, the idea of Python 3000 was born. We had so many things that we wanted to fix, and we knew how to fix them. But the fix wouldn't be backward compatible, so we decided to create a new version of the language that fixed a whole bunch of those things while avoiding making the new version of the language so different that it would alienate the users.

Python 3000 (or “py3k”) became Python 3, and the Python community took a very patient approach to the multiyear transition. They knew it would be necessary to fully support both Python 2 and Python 3 for many years. Python 3 was released in December 2008 and even now, seven years later, there is still strong and active support for Python 2. Python's leadership is letting the community decide when to make the switch:

Every year at PyCon [the largest annual gathering of the Python community] I get a sense of who is using Python 3 and how

happy they are. I see tremendous progress every year, but we're still not where we want to be. At the 2014 Python conference, I had to hold off a whole bunch of people with very good arguments as to why we should do a Python 2.8 release. Python 3 is winning the race but it still needs more time.

Python 3 is certainly past the tipping point and gaining more inertia all the time. But the approach to a version transition that has spanned nearly a decade of planning, development, and adjustment is a testament to letting an open community design its own destiny.

With the influx of data mining and data analysis, Python is well positioned to become the premier language of data analysis and the language of choice for new programmers. The Python community is strong and active—PyCon now attracts more than 2,500 attendees annually, and there are more than 40 Python-related conferences each year in various locations around the world.

Python is also a study of how the leadership of a community can sustain

itself by working for open-minded companies that allow their employees to participate in building an open technology commons. Over the past two decades, Guido has worked for many companies that allowed him to remain involved with Python, including Centrum Wiskunde & Informatica, CNRI, Zope, Google, and now Dropbox. Guido brought the skills and knowledge he gained as an open source leader to each company, and in turn, the companies supported his involvement with Python. It's a great model for sustainable open source technology. **■**

CHARLES SEVERANCE, Computing Conversations column editor and *Computer's* multimedia editor in chief, is a clinical associate professor in the School of Information at the University of Michigan. Follow him on Twitter @drchuck or contact him at csev@umich.edu.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.



Subscribe today!

IEEE Computer Society's newest magazine tackles the emerging technology of cloud computing.

[computer.org/
cloudcomputing](http://computer.org/cloudcomputing)

IEEE  computer society

 IEEE COMMUNICATIONS SOCIETY