David Kocen
Professor Barrett
CSU33012
18 October 2019

Software Engineer Biography: Guido van Rossum

Biography

Guido van Rossum is a software engineer best known for the creation of Python. Originally from the Netherlands, Van Rossum studied mathematics and computer science at the University of Amsterdam. In a 2016 speech to celebrate King's Day, Van Rossum described himself as "a late bloomer. I graduated from college when I was 26. I was 45 when I got married. I'm now 60 years old, with a 14 year old son...I've lived in the US for over 20 years and I am still a permanent resident" (Van Rossum, 2016). After graduating, Van Rossum went to work for the Centrum Wiskunde & Informatica on a research team building a new programming language called ABC. The goal of ABC was to "stamp out Basic" (Van Rossum, 2016). Though ABC would fail, its philosophy of being "a language designed to be easy to use for scientists without a programming background" (Hsu, 2018) would heavily influence Van Rossum's later work on Python.

After ABC's failure, Van Rossum worked on Amoeba which had the goal of turning "a collection of workstations...into a transparent distributed system" (Vrije Universiteit, 1996). Sick of wasting time writing small functions in C, Van Rossum decided to write a language "that would be simple and convenient for quick and dirty tasks...but with the expressivity of a high-level programming language" (Hsu, 2018). This language would become Python. From the start, Van Rossum was committed to keeping Python open-source leading to the creation of the Python Software Foundation. The PSF's goal is the "advancing [of] open source technology related to the Python programming language"(Python Software Foundation, 2009). Through the proposal and adoption of Python Enhancement Proposals, or PEPs, community involvement lies at the core of Python's development (Warsaw et. al., 2000). In the late 1990s, the PSF appointed Van Rossun the position of Benevolent Dictator for Life.

Van Rossum immigrated to the United States in 1994 where he has since held a number of jobs working for the National Institute of Standards and Technology, the Corporation for National Research Initiatives, Zope, Google, and currently Dropbox. Throughout this time he continued to develop and promote Python (Hsu, 2018). On July 12, 2018 Van Rossum officially stepped down from his position as Benevolent Dictator for Life citing the desire to not "want to have to fight so hard for a PEP and find that so many people despise [his] decisions" (Van Rossum, 2018). Despite this, he remains very active in the Python community frequently attending PyCon conventions and advocating for the use of Python in both a professional and educational setting (Van Rossum, 2016).

Personal Significance

Guido van Rossum's impact in the field of software engineering is quite apparent given that Python is currently the third most used language on GitHub (GitHub Inc., 2018). However, Van Rossum is of particular significance to me because of his commitment to intelligent programming. In addition to computer science, I study psychology. Because of this I am particularly interested in the psychology of programming. The psychology of programming explores a wide range of psychological topics and applies them to writing code. This includes ideas such as entering a state of flow when programming, understanding how the mind represents abstract concepts, and improving the readability and understanding of code. I would argue that Van Rossum, whether aware of the underlying psychological principles or not, has had a huge impact on the development efficiency because of his commitment to creating, with the help of an open source community, a simple, readable, and highly expressive language.

Within the Python community is the idea of writing Pythonic code. Summarized in PEP 20, "The Zen of Python", Pythonic code is code that is beautiful, explicit, simple, flat as opposed to nested, and readability focused (Peters, 2004). A great example of this is the use of list comprehensions to create lists using pre-existing lists in one succinct line of code rather than a for loop. For myself, the first time I really understood what was meant by beautiful code was a simple Python trick for swapping variable values in one line. Van Rossum understands that programming languages, like any other language, are "how programmers express and

communicate ideas" (Van Rossum, 2016). He developed Python with the human factor in mind. "The computer can take care of itself" (Van Rossum, 2016) but to truly be effective, a programmer must be able to express their ideas to other programmers. Van Rossum had found a way to do this without sacrificing functionality. Interestingly, he attributes Python's success to being "developed on the Internet, entirely in the open" (Van Rossum, 2016) suggesting an innate understanding of the wisdom of crowds. By enthusiastically leading a collaborative team of volunteers, Van Rossum has created a way for people to clearly communicate abstract ideas in order to develop incredible pieces of software.

I personally believe that everyone should learn to program. This is not because everyone should develop software but because learning to program is learning to think about a problem systematically in order to find an optimal solution. However, programming by its very nature is hard. Humans are not good with abstract concepts and explaining step-by-step processes. Van Rossum acknowledged the importance of learning to program back in 1999 with his DARPA proposal Computer Programming for Everybody. He believed that "virtually everybody can obtain some level of computer programming skills in school"(Van Rossum, 1999) and so went about creating a language that could make this a reality. Python is now used in many introductory level computer science courses. Rather than getting bogged down with unintuitive syntax and processes, Van Rossum focused on enabling students to get at the heart of programming: turning a logic problem into an elegant solution. The already hard task of learning to program is made easier because of these efforts.

Learning to code is more accessible than it has ever been thanks largely to the work of Guido van Rossum. At the start of developing Python he knew that he wanted to create a language that allowed for the clear, efficient communication of ideas without sacrificing functionality. By keeping Python open sourced Van Rossum has become a respected figure in the open source movement. He has intuitively applied key concepts in psychology to the development of better code and his work has allowed me to see clearly the underlying beauty of program development.

References

Github Inc. (2018). The State of the Octoverse. Retrieved October 13, 2019 from https://octoverse.github.com/

Hsu, H. (2018). 2018 Museum Fellow Guido van Rossum, Python Creator & Benevolent Dictator for Life. *CHM Blog*. Retrieved October 13, 2019 from https://computerhistory.org/blog/2018-chm-fellow-guido-van-rossum-python-creator-benevolent-dictator-for-life/

Peters, T. (2004). PEP 20--The Zen of Python. *Python Developer's Guide.* Retrieved October 13, 2019 from https://www.python.org/dev/peps/pep-0020/

Python Software Foundation (2009). Python Software Foundation Mission Statement. *Python Software Foundation.* Retrieved October 13, 2019 from https://www.python.org/psf/mission/

Van Rossum, G. (1999). Computer Programming for Everybody (Revised Proposal): A Scouting Expedition for the Programmers of Tomorrow. Retrieved October 13, 2019 from https://www.python.org/doc/essays/cp4e/

Van Rossum, G. (2016). King's Day Speech. *Neopythonic*. Retrieved October 13, 2019 from http://neopythonic.blogspot.com/2016/04/kings-day-speech.html

Van Rossum, G. (2018). Transfer of power. *python-commiters Mailing List*. Retrieved October 13, 2019 from https://mail.python.org/pipermail/python-committers/2018-July/005664.html

Vrije Universiteit. (1996). The Amoeba Distributed Operating System. Retrieved October 13, 2019 from https://www.cs.vu.nl/pub/amoeba/amoeba.html

Warsaw, B., Hylton, J., Goodger, D., & Coghlan, N. (2000). PEP 1--PEP Purpose and Guidelines. *Python Developer's Guide.* Retrieved October 13, 2019 from https://www.python.org/dev/peps/pep-0001/