

Python Programming, Rust and Puppet Enterprise 3

By *Roy Schestowitz*

Created *13/12/2019 - 2:47am*

Submitted by Roy Schestowitz on Friday 13th of December 2019 02:47:30 AM Filed under [Development](#) [1]

- [Circuit Python at PyConf Hyderabad](#) [2]

Coding in/with hardware has become my biggest stress buster for me ever since I have been introduced to it in PyCon Pune 2017 by John. Coding with hardware provides a real-life interaction with the code you write. It flourishes creativity. I can do all of this while I learn something new. Now I look for auctions to offer me a chance to code in/with Hardware. It gives the chance to escape the muggle world.

- [New in testmon 1.0.0](#) [3]

Significant portions of testmon have been rewritten for v 1.0.1. Although the UI is mostly the same, there are some significant differences.

- [Determining affected tests](#) [4]

Automatically determining affected tests sounds too good to be true. Python developers rightfully have a suspecting attitude towards any tool which tries to be too clever about their source code. Code completion and symbol searching doesn't need to be 100% reliable but messing with the test suite execution? This page explains what testmon tries and what it does not try to achieve.

[...]

After running the test with coverage analysis and parsing the source code, testmon determines which blocks does `test_s.py::test_add` depend on. In our example it's Block 1,2 and 4. (and not

Block 3). testmon doesn't store the whole code of the block but just a checksum of it. Block 3 can be changed to anything. As long as the Block 1,2 and 4 stay the same, the execution path for test_s.py::test_add and it's outcome will stay the same.

- [How to set-up and use py.test in Pycharm](#) [5]

I've been using Vim and terminal as a weapon of choice for years. I've had a good time with it, however, more and more people ask me why I'm using this setup. And honestly, I don't know the answer.

I'm aware that things can be done more efficiently and an IDE can help with a lot of things. I guess that my weak spot is the unit tests and testing my code in general. I'm not running my tests when on the coding spree, I'm breaking lots of stuff, and only when I think I'm finished, I'll do the fixing and make everything running green again.

Well, I would like to change that. And I'm also curious about trying out new ways of doing things. The obvious choice for programming in Python is the PyCharm. It's a nice IDE, supports many features that I like and most importantly, it can help with the testing. PyCharm can easily integrate with popular test frameworks and run the tests for me.

- [This Week in Rust 316](#) [6]

- [Continuous Delivery for Puppet Enterprise 3.0 is now available](#) [7]

I am very excited to announce the immediate availability of Continuous Delivery for Puppet Enterprise 3.0! Over the last year, we've taken to heart the challenges and recommendations our customers have shared with us on how we can make Continuous Delivery for Puppet Enterprise better. Our intent is to be truly customer-obsessed, meet our customers where they are, and help them get to where they want to be. This release focuses on our customers' needs by providing more context into the impact of a proposed Puppet change by offering Hiera support for Impact Analysis, a simplified approach to defining pipelines as code, and the ability to easily compose custom deployment processes (currently in beta!). Let's dive in!

[Development](#)

Source URL: <http://www.tuxmachines.org/node/131645>

Links:

[1] <http://www.tuxmachines.org/taxonomy/term/145>

[2] <https://anweshadas.in/circuit-python-at-pyconf-hyd/>

[3] <https://testmon.org/new-in-testmon-100.html>

[4] <https://testmon.org/determining-affected-tests.html>

[5] <https://testmon.org/how-to-set-up-and-use-pytest-in-pycharm.html>

[6] <https://this-week-in-rust.org/blog/2019/12/10/this-week-in-rust-316/>

[7] <https://puppet.com/blog/continuous-delivery-for-puppet-enterprise-3-0-is-now-available/>