Software Engineering Conference Russia
October 2017, St. Petersburg

On development of a framework for massive source code analysis using static code analyzers

Alex Chistyakov, ITMO University

What is it all about?

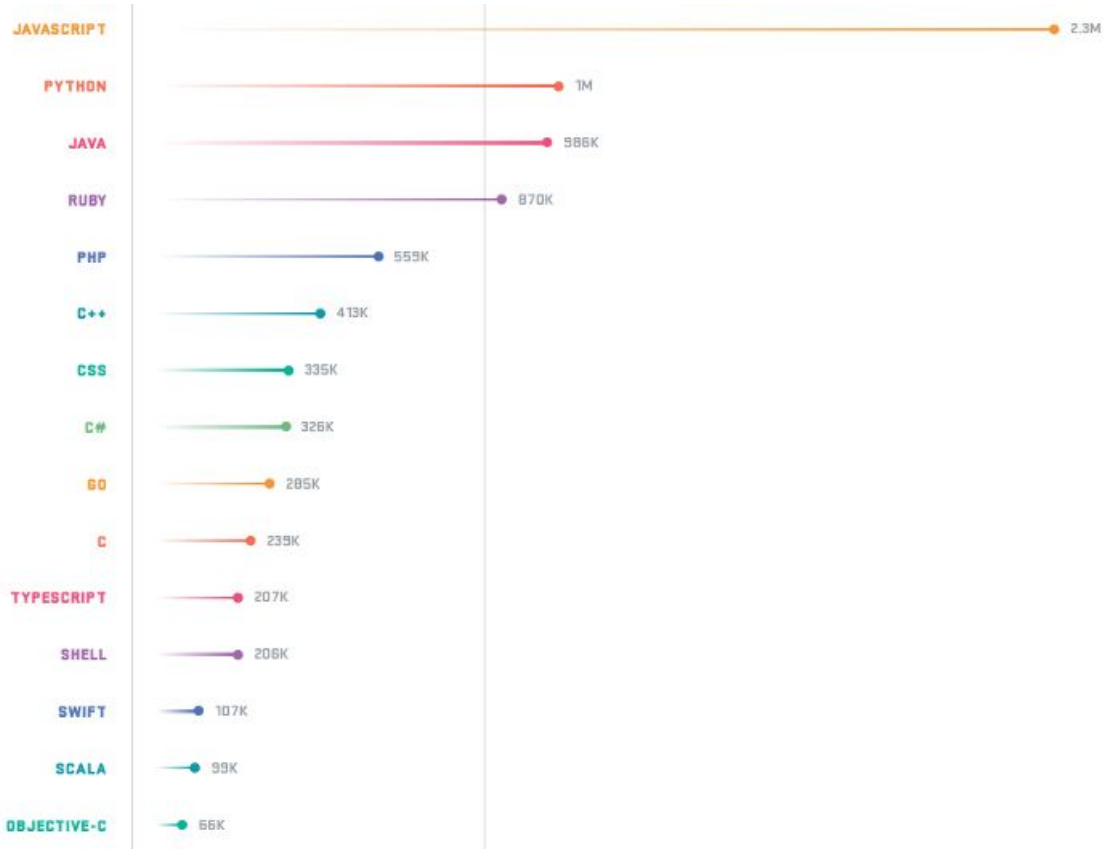
- Dynamic programming languages prevail

What is it all about?

- Dynamic programming languages prevail
- Python and JS are the most common ones

What is it all about?

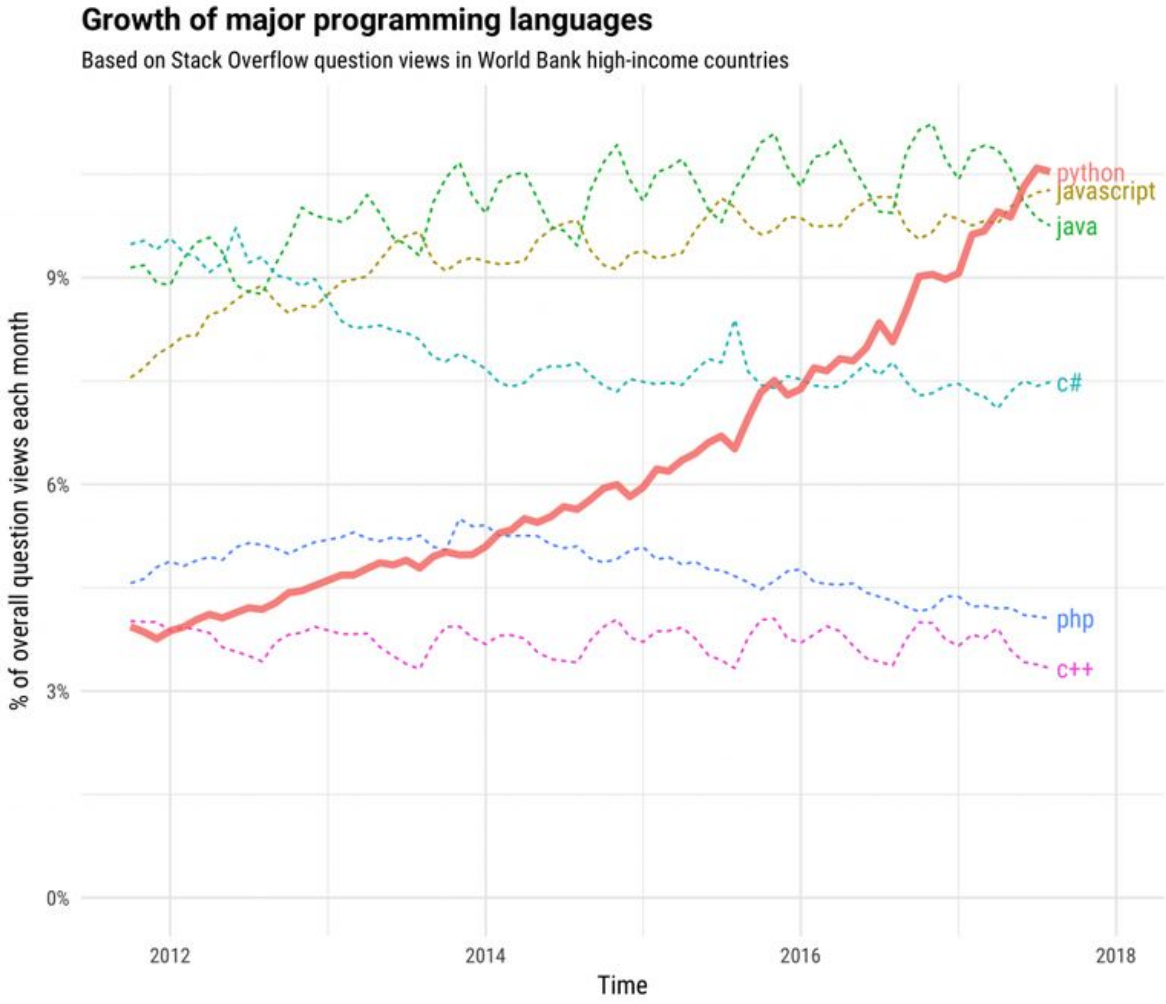
- Source: Github stats for 2017



What is it all about?

- Dynamic programming languages prevail
- Python and JS are the most common ones
- Python market share is the fastest growing

What is it all about?



What is code quality?

- For practical reasons: number of error in code

What is code quality?

- For practical reasons: number of error in code
- How do we estimate the number of errors in code?

What is code quality?

- For practical reasons: number of error in code
- How do we estimate the number of errors in code?
- We don’t

What is code quality?

- For practical reasons: number of error in code
- How do we estimate the number of errors in code?
- We don’t
- Number of resolved bugs in a tracker is what matters

Do static analyzers help?

- They exist for that reason

Do static analyzers help?

- They exist for that reason
- Let’s gain some statistics

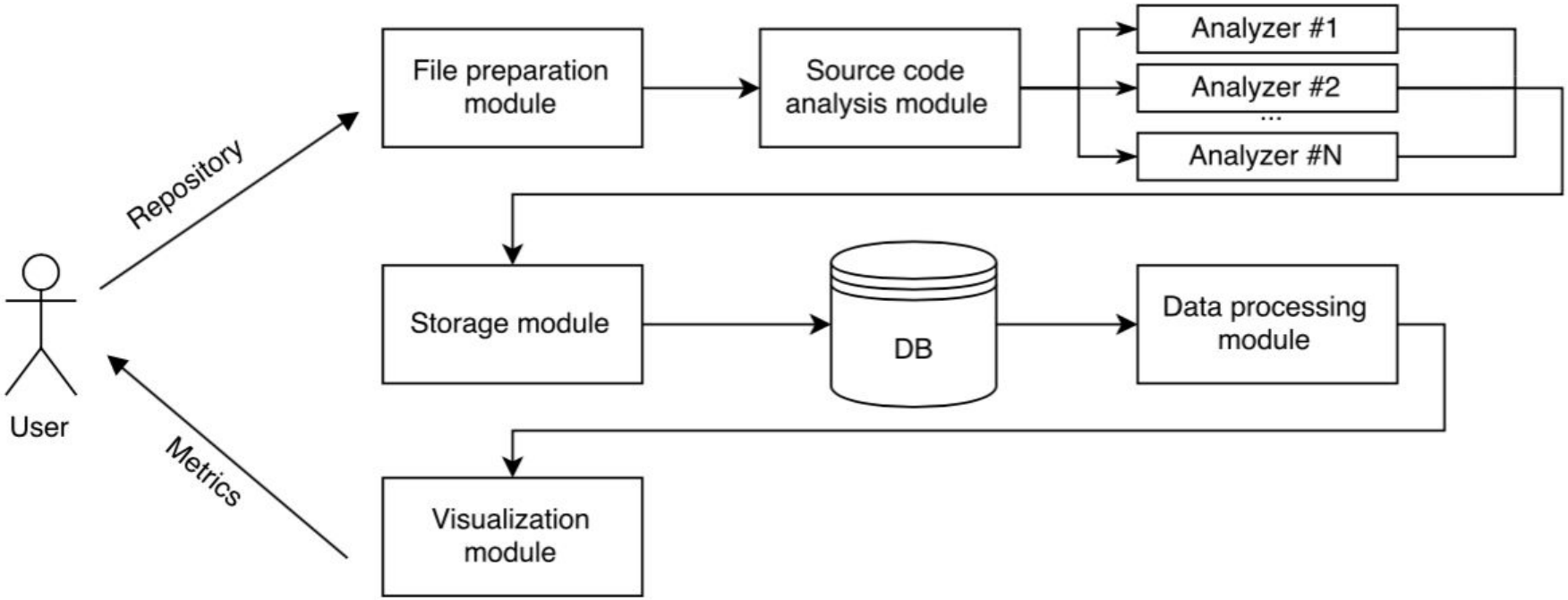
Do static analyzers help?

- They exist for that reason
- Let’s gain some statistics
- Is there a correlation between number of closed defects in a tracker and number of warnings produced by a static analyzer?

Let’s check!

- Let’s grab a number of popular static analyzers
- Let’s analyze top Python repos from Github
- Let’s see if number of warnings produced by analyzers relate to number of defects in a project bug tracker

High-level architecture



SECR

Static analyzers for Python

- Coala
- Pylama
- Flake8
- Some other products

Coala

- Supports plugins
- Is written in Python

Pylama

- Wraps a number of other analyzing tools
- Is written in Python

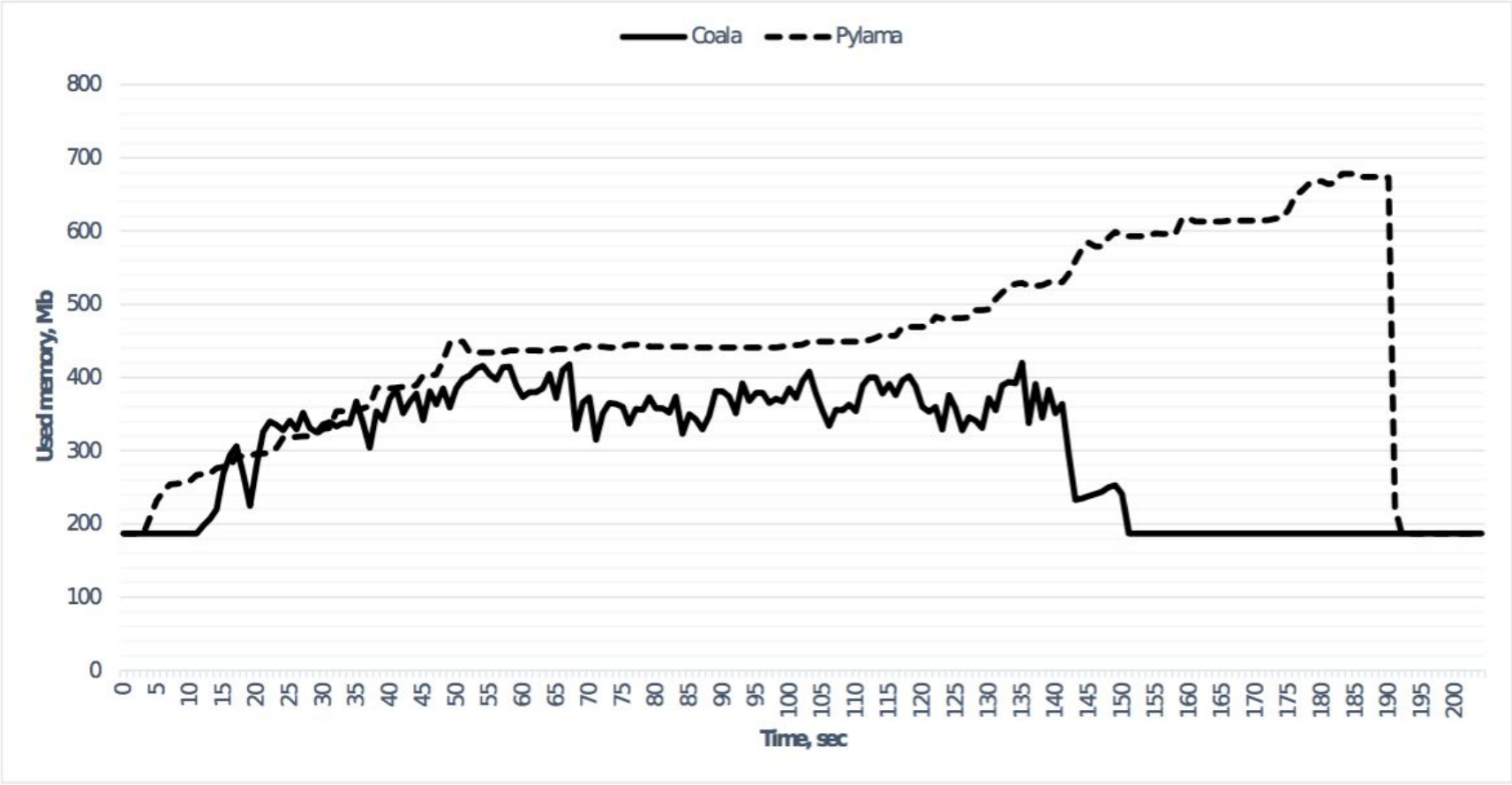
Flake8

- Enforces coding style
- Wraps three other tools
- Is written in Python

Ready, set, go!

- We chose Coala and Pylama because Flake8 does not pretend to be a static analyzer
- We cloned some top Python repos from Github (namely, Ansible, Coala itself and some others) and started to analyze

Coala suddenly wins!



SECR

Results

- There is no visible correlation between number of closed bugs in tracker and results of code analyzing (we kind of expected that)

Results

- There is no visible correlation between number of closed bugs in tracker and results of code analyzing (we kind of expected that)
- We were unable to find an open source Python code analyzer which builds an AST prior to perform analyzing

Related work

- “A Large-Scale Study of Programming Languages and Code Quality in GitHub” by Baishakhi Ray, Daryl Posnett, Premkumar Devanbu, Vladimir Filkov published Oct, 2017

Related work

- “A Large-Scale Study of Programming Languages and Code Quality in GitHub” by Baishakhi Ray, Daryl Posnett, Premkumar Devanbu, Vladimir Filkov published Oct, 2017
- Python is not the best choice
- Functional programming matters
- It’s hard to measure an impact of language itself

Conclusion

- Open source static code analyzers for Python are mostly useless
- There is no correlation between number of defects in a project tracker and number of problems detected by a static code analyzer
- This is a work in progress, more results to follow

Questions?

- Irina Radchenko, associate professor, ITMO University, iradche@gmail.com
- Artem Pripadchev, graduate student, ITMO University, getviruses@gmail.com
- Alex Chistyakov, researcher, ITMO University, alexclear@gmail.com