

Continuous Updating

How do you keep track
of your **LIBRARIES?**

How many **DEPENDENCIES** do
you have in your project?

Which **LICENSES** are
your dependencies
using?

You don't know?

Goldman Sachs sent a brilliant computer scientist to **JAIL!**



GPL License

<http://blog.garrytan.com/goldman-sachs-sent-a-brilliant-computer-scientist-to-jail-over-8mb-of-open-source-code-uploaded-to-an-svn-repo>

15 years ago we used to work with the **WATERFALL MODEL**

A diagram of the Waterfall Model overlaid on a photograph of a waterfall. The waterfall flows over mossy rocks, creating a series of small cascades. The water is white and frothy, contrasting with the dark, moss-covered rocks. The background is a lush green forest.

Requirements



Analysis



Design



Coding



Testing



Accepting

But today we are **AGILE!**



Everything the Waterfall Model used
to execute in one year ...

... we nowadays execute in 2 weeks!



The way we develop software
today totally changed!

Being **AGILE** got us



CONTINUOUS Refactoring

CONTINUOUS Testing

CONTINUOUS Integration

CONTINUOUS Delivery

But what about

CONTINUOUS Updating ?

How do you update your
LIBRARIES ?

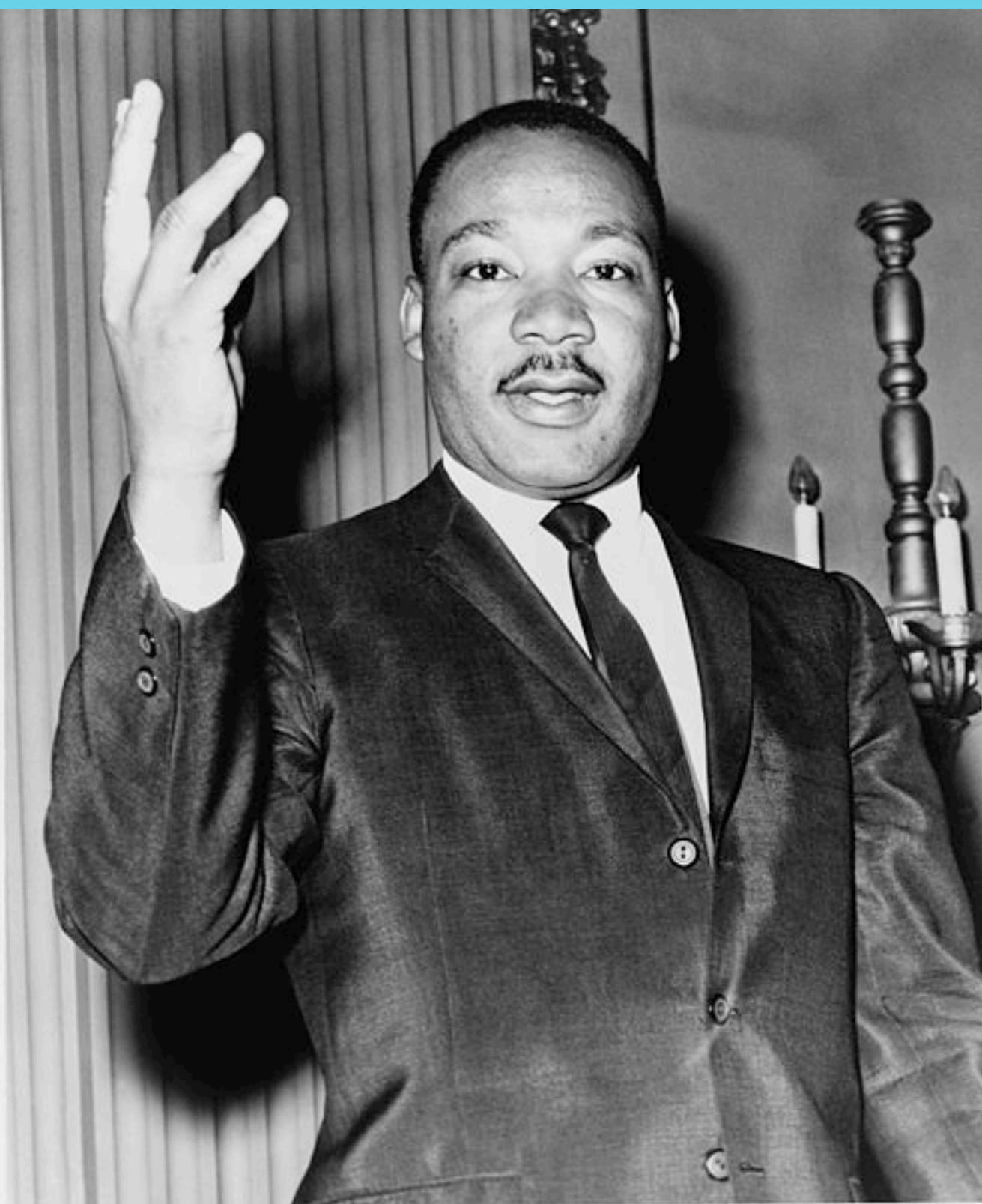
You don't ?

Because you never touch a
running system, right?

```
//COBUCLG JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//HELOWRLD EXEC COBUCLG,PARM.COB='MAP,LIST,LET'
//COB.SYSIN DD *
001 IDENTIFICATION DIVISION.
002 PROGRAM-ID. 'HELLO'.
003 ENVIRONMENT DIVISION.
004 CONFIGURATION SECTION.
005 SOURCE-COMPUTER. IBM-360.
006 OBJECT-COMPUTER. IBM-360.
0065 SPECIAL-NAMES.
0066     CONSOLE IS CNSL.
007 DATA DIVISION.
008 WORKING-STORAGE SECTION.
009 77 HELLO-CONST PIC X(12) VALUE 'HELLO, WORLD'.
075 PROCEDURE DIVISION.
090 000-DISPLAY.
100     DISPLAY HELLO-CONST UPON CNSL.
110     STOP RUN.
//LKED.SYSLIB DD DSNAME=SYS1.COBLIB,DISP=SHR
//          DD DSNAME=SYS1.LINKLIB,DISP=SHR
//GO.SYSPRINT DD SYSOUT=A
//
```

So you wanna work
with **COBOL**? Right?

Enjoy!



“If you can't fly then run, if you can't run then walk, if you can't walk then crawl, but whatever you do you have to keep moving forward.”

Martin Luther King Jr.

Core committers don't release
new versions just for fun!

They always have good reasons



- Bug Fixes
- Security Fixes
- Speed & Memory optimization
- New Features

How do you ensure that new versions don't break the system?

Semantic Versioning

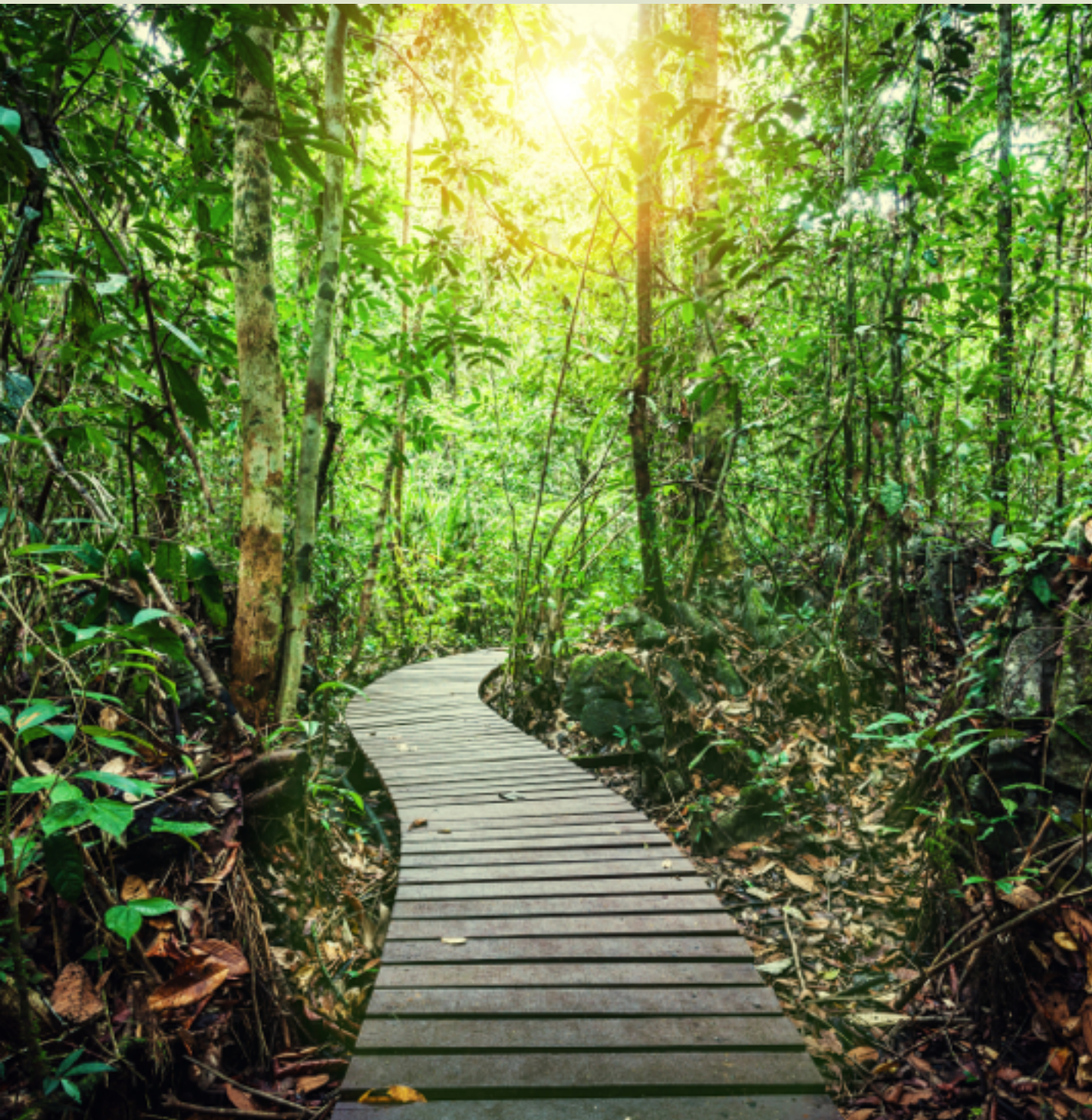
Migration Paths

Continuous Testing

<http://semver.org/>

MAJOR.MINOR.PATCH

1. MAJOR version when you make incompatible API changes
2. MINOR version when you add functionality in a backwards-compatible manner
3. PATCH version when you make backwards-compatible bug fixes.



Always follow the
MIGRATION PATH!

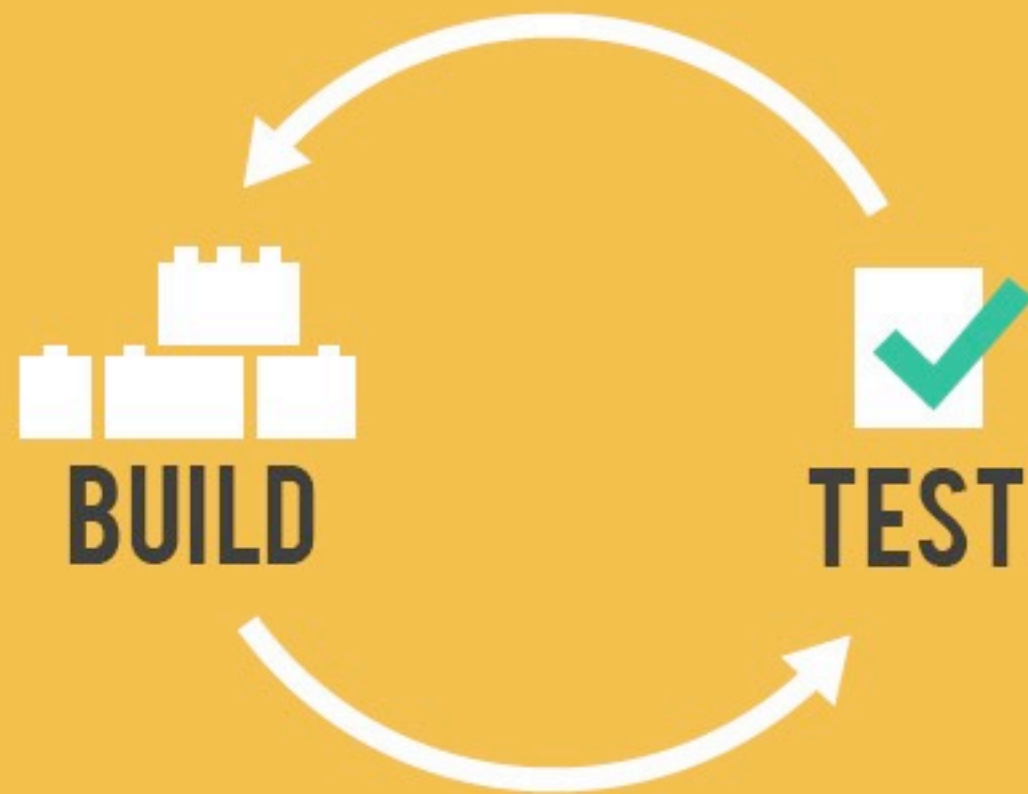


Many small steps are better than one big step

You can do **SMALL MIGRATIONS** on the fly.

BIG MIGRATIONS are risky and expensive.

If you miss versions, you miss migration paths, too.
And that leads to **TROUBLE!**



Always run your **TESTS** against new versions

Another reason for being current



Do you really believe
those young talents
wanna work with
COBOL?
Or other **OLD SHIT?**

Tracking versions is a pain!



**SOFTWARE LIBRARIES
are NOT like iPhone
Apps!**



**100 libraries per
project in avg.**

After 2 weeks the first libraries are OUT-DATED!

Developers are missing critical **BUB FIXES**
and important **UPDATES!**



Manually checking for updates is no fun!

It cost **TIME & MONEY!**

NOBODY WANTS TO DO IT!

So, how do you wanna solve this
PROBLEM?

You have to
AUTOMATE!

You need a **TOOL** for that!

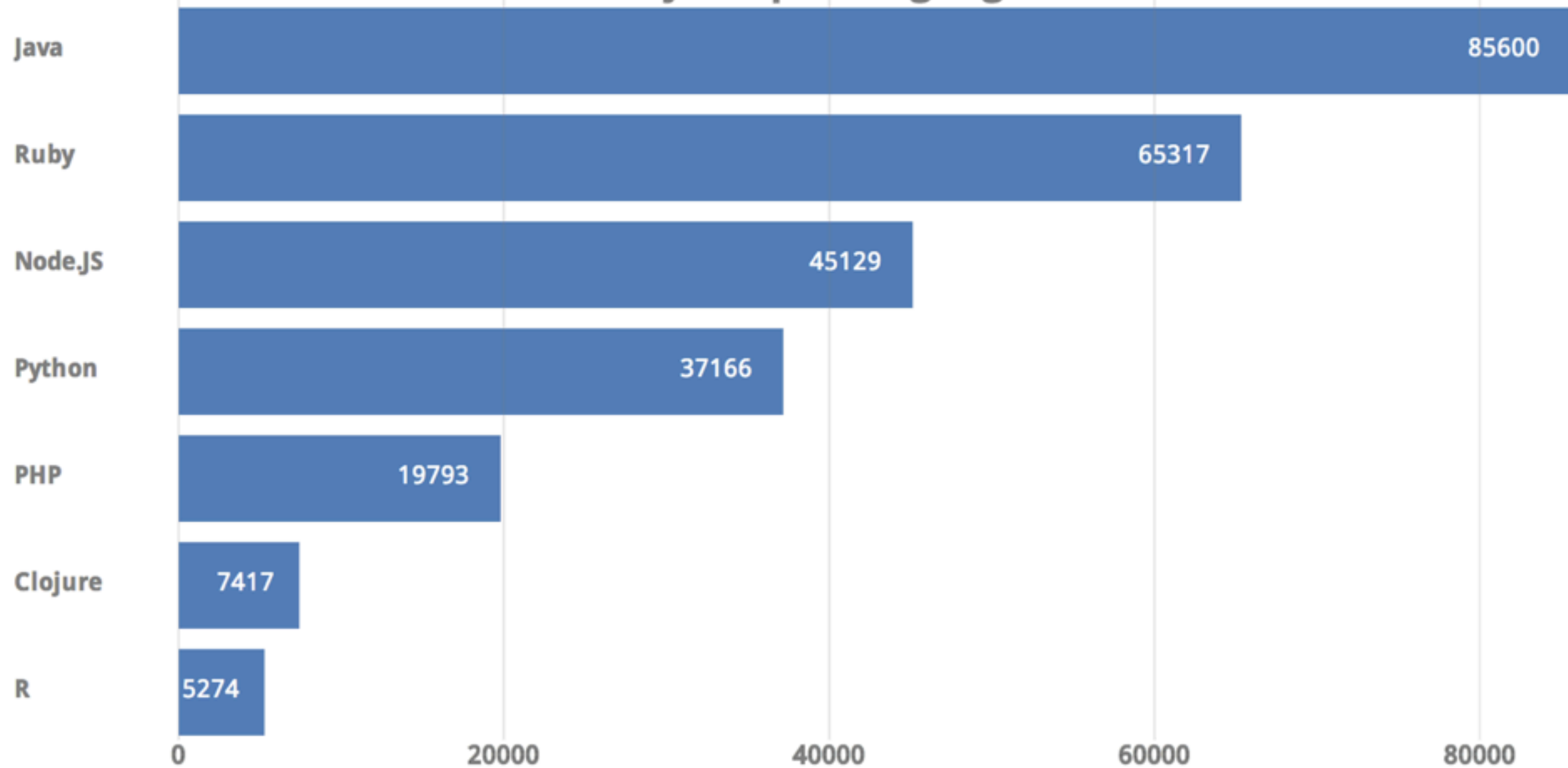
	VersionEye	Gemnasium	GemNotifier
Languages	Java, Ruby, Node.JS, Python, PHP, Clojure, R, JavaScript	Ruby, Node.JS	Ruby
Project Integration	GitHub, URL, FileUpload, API	GitHub	Single Subscribe
Changelogs	in progress	yes	no
Security	in progress	yes	no
Licenses	yes	no	no
API	yes	no	no

www.VersionEye.com

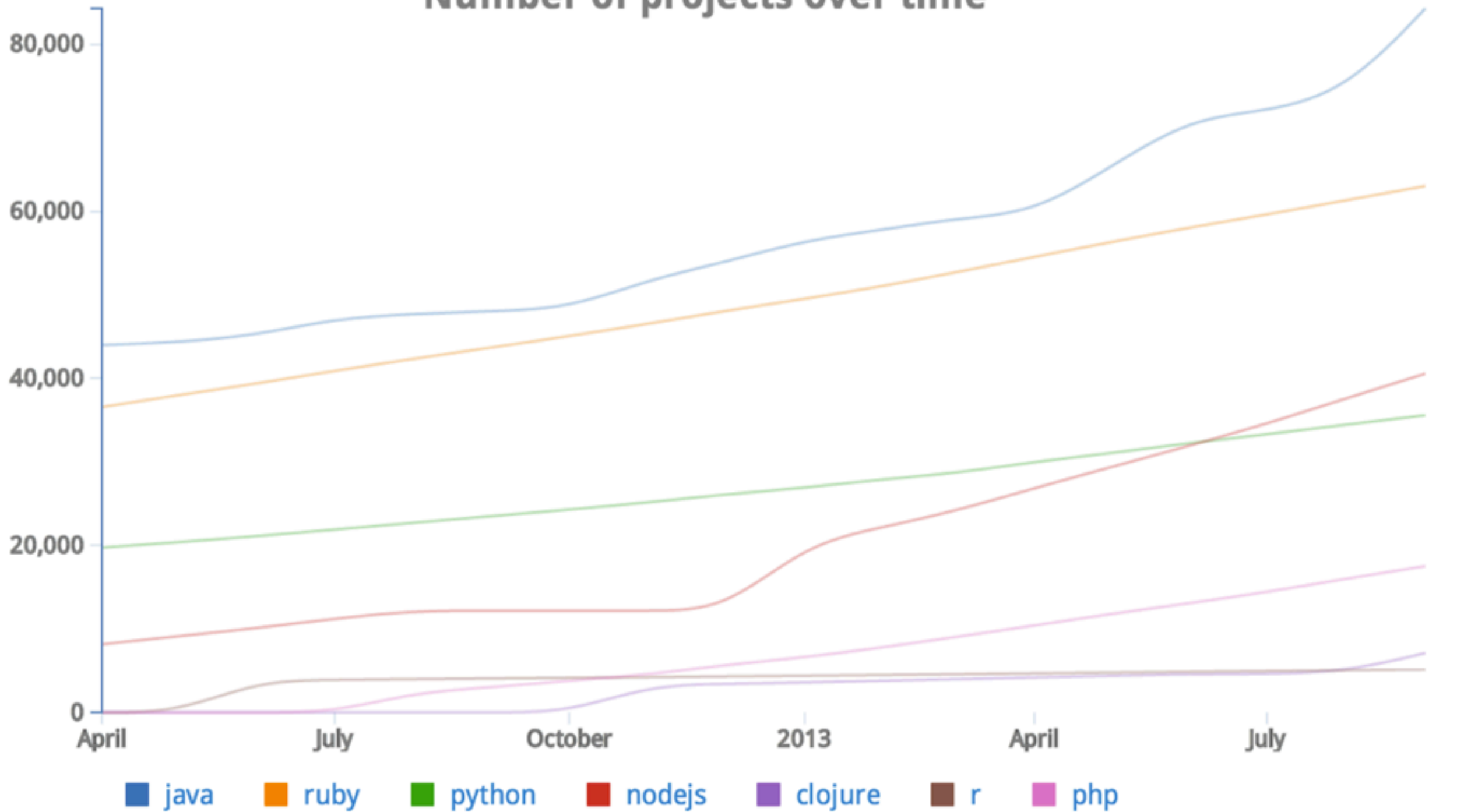
Keeps an eye on more than 250K open source libraries!

Supports 8 Languages and 7 Package Managers!

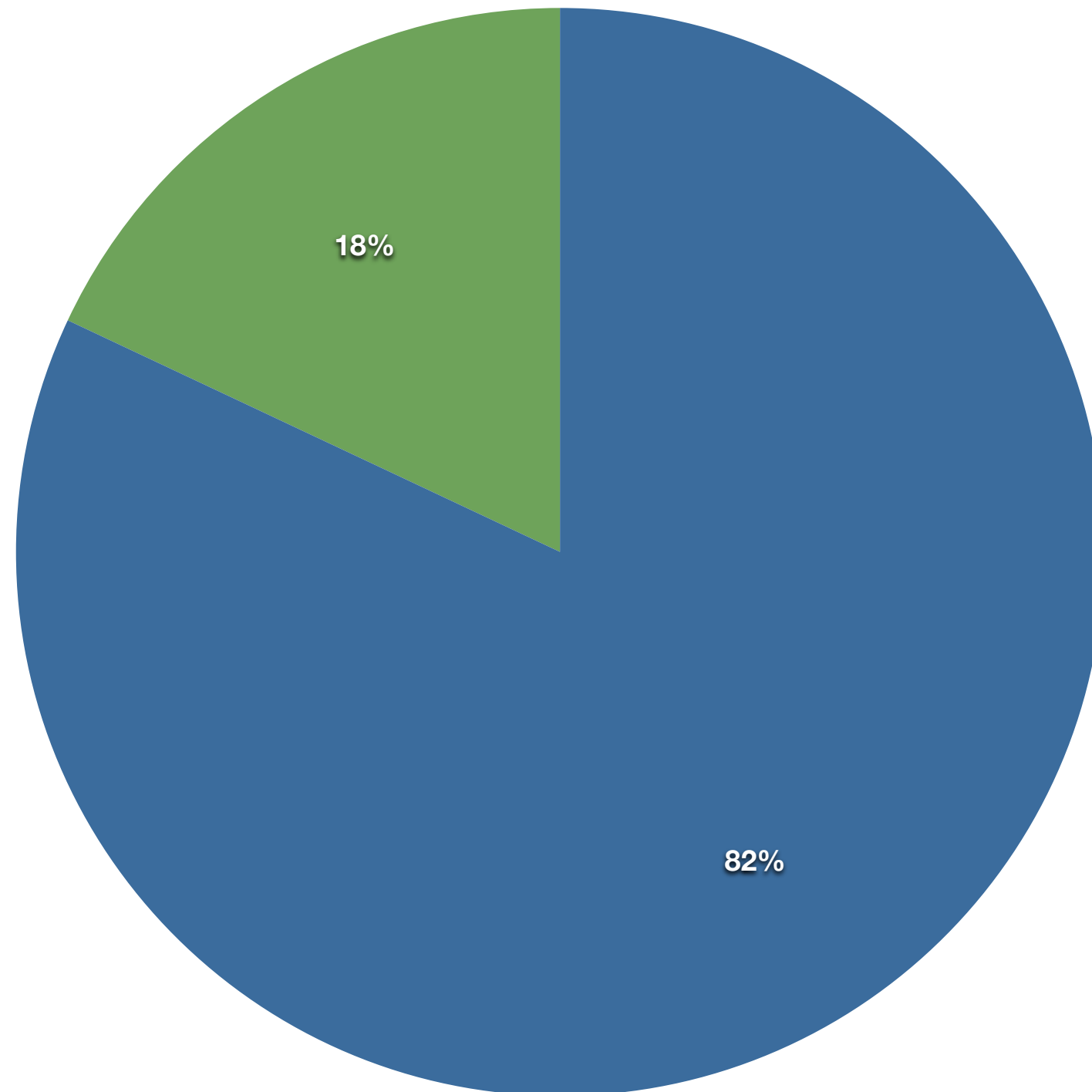
Projects per language



Number of projects over time



Java Open Source Libraries



● Central MVN Repo ● Other Repos



QUESTIONS?

**Contact me
on Twitter**

@RobertReiz