Analyzing Open-source Scientific Software Projects

Kanika Sood[1], Anshu Dubey[2], Boyana Norris[1], Lois Curfman McInnes[2]
University of Oregon[1], Argonne National Laboratory[2]

Problem

Traditional

- The productivity of scientific software developers typically gets less attention than application performance or publications
- Standard software metrics provide limited insight
- Higher project complexity → difficult to estimate individual/team productivity and project maturity/acceptance

Outcome

- Limited software productivity metrics → lost opportunity to reduce cost and increase scientific output
- Limited metrics on project maturity and audience acceptance → limited future project planning and funding projections

Our Approach

Our approach aims to understand patterns in software development and capture productivity metrics.

Step 1

- Understand and analyze diverse attributes that impact productivity and can be captured across a broad range of projects in high-performance computational science.

Step 2

- Identify characteristics that form *dominant* attributes across various projects and use them to produce metrics for insight into *productivity improvement or degradation.*

Step 3

- Test the metrics on selected projects to gain insight into project productivity, team productivity, project success, and project maturity.

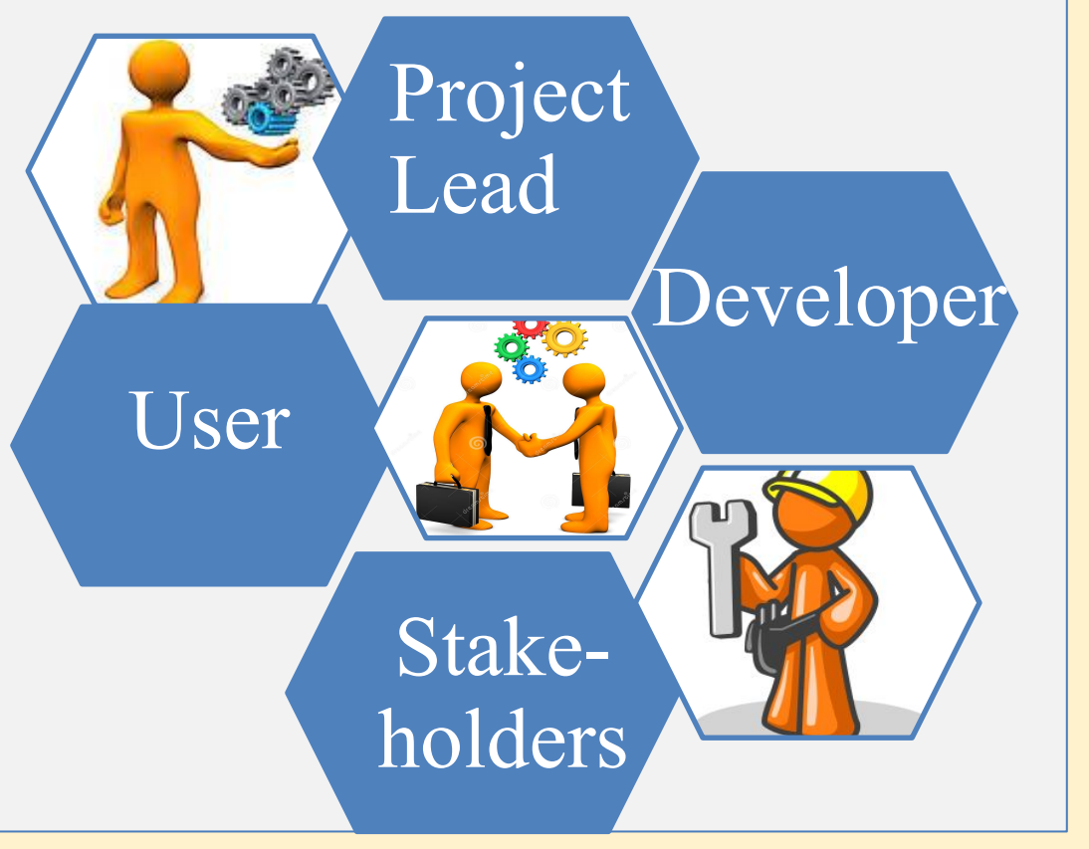
Productivity Metrics

1. **Project health**
   a. **Community interest**: Identify how interested the community is based on the number of issues created (bugs, enhancements, others) on a monthly basis.
   b. **Stability**: Indicate project stability in terms of support provided for users based on issue requests and issue categories.
   c. **Growth**: Analyze project growth with respect to code size and complexity.
2. **Code analysis**
   a. **NLOC**: Examine the change in total number of lines of code for a project in its lifetime.
   b. **Size**: Evaluate the number of code files for a project in its lifetime.
   c. **Complexity**: Determine the average cyclomatic complexity (CCN) for project releases. [CCN: quantitative measure of the number of independent control-flow paths in code.]
3. **Developer participation**
   a. **Activity**: Determine developer participation based on pull requests.
   b. **Contributions**: Compute the percentage of code files touched by an individual.
   c. **Reliance**: Analyze project reliance on individual developers.

Conclusions

We analyze HPC software repositories and compute metrics that can be used to better understand and potentially improve software development practices. We derive metrics that provide insight into development practices, project growth, and overall project health.

We also perform fine-grain analysis for individual code files, patterns in software trends, and individual developer efforts. We study the impact of code growth and changes in code complexity over project lifetimes.



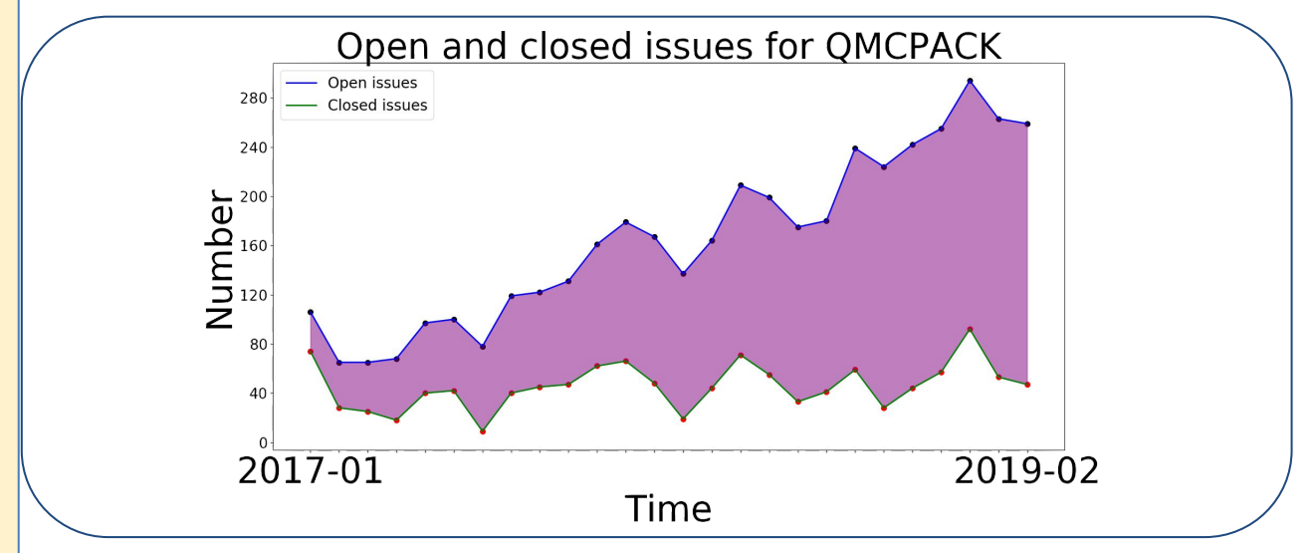
Acknowledgments

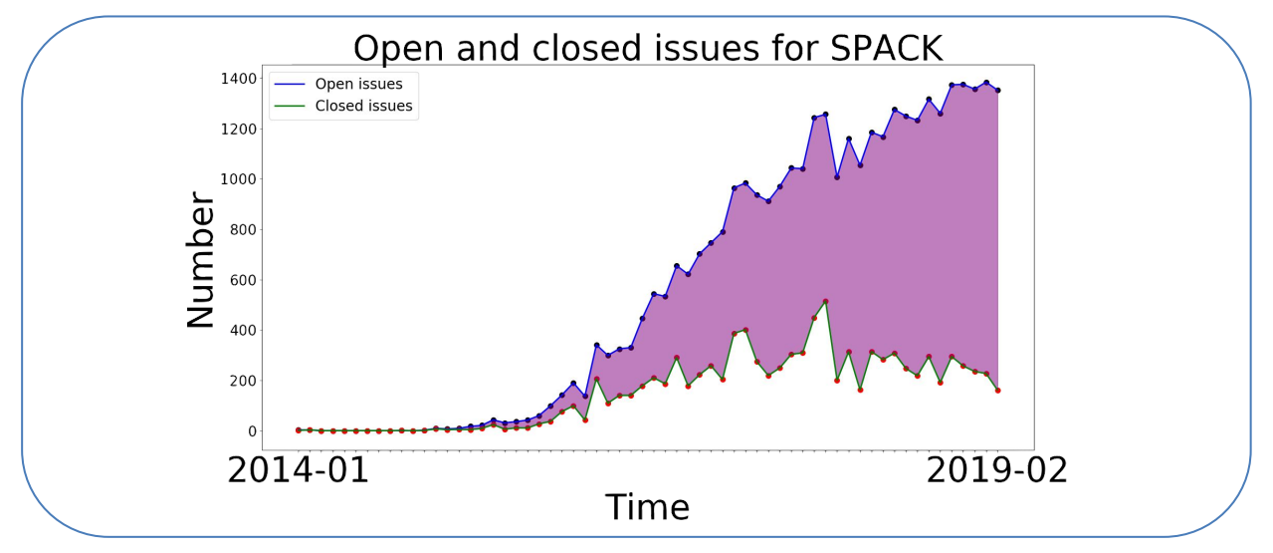
This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

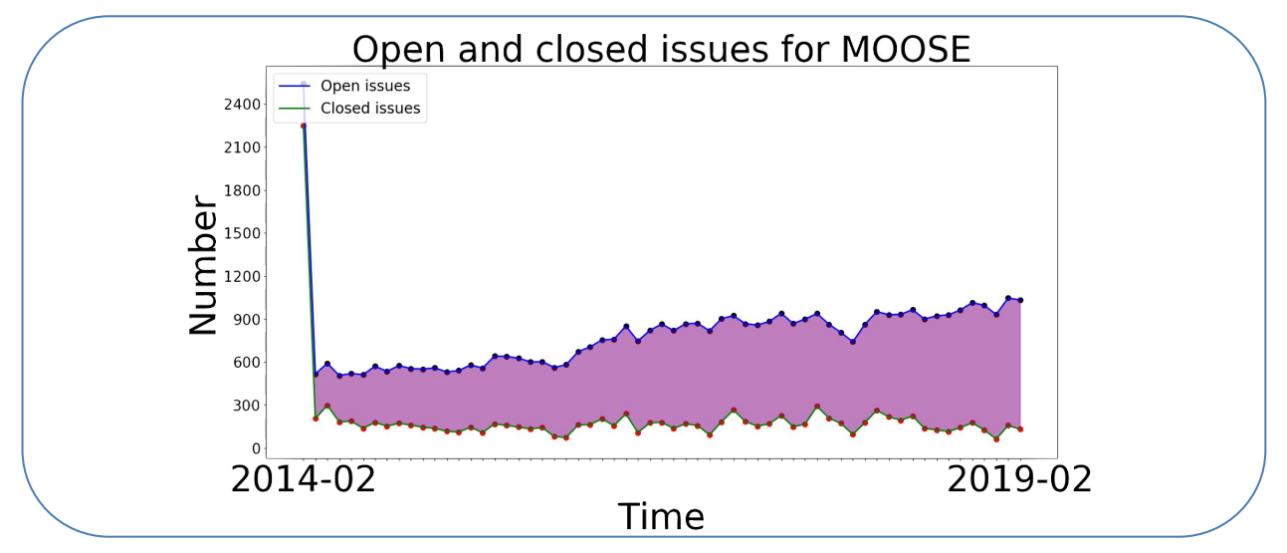
Observations and Insights

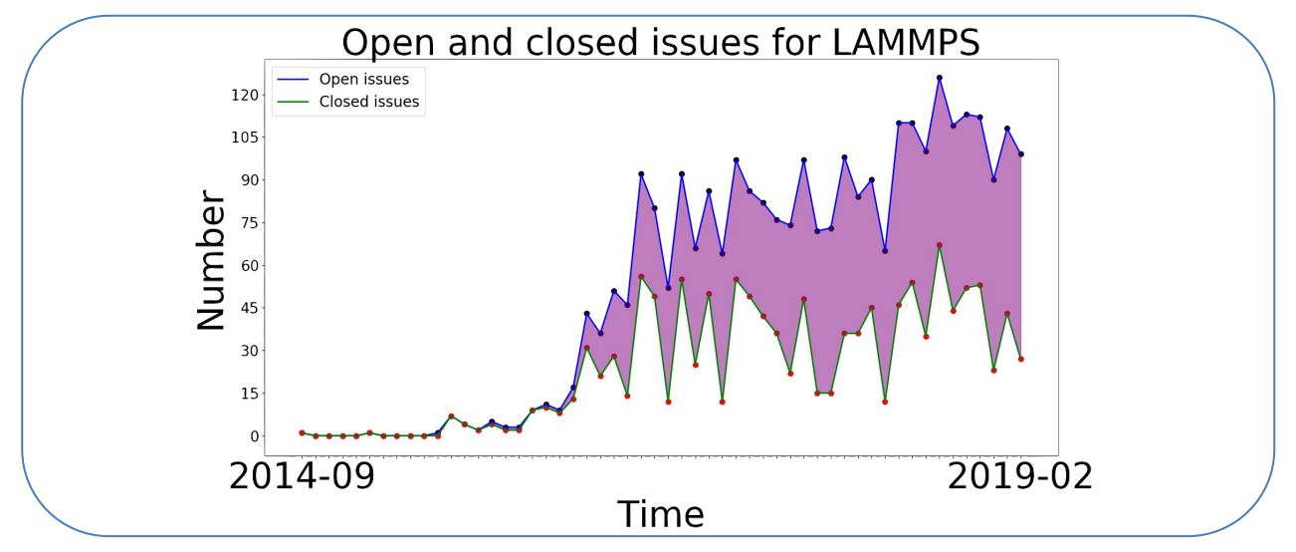
QMCPACK | SPACK | MOOSE | LAMMPS | YT | NWCHEM | QDPXX | GROMACS

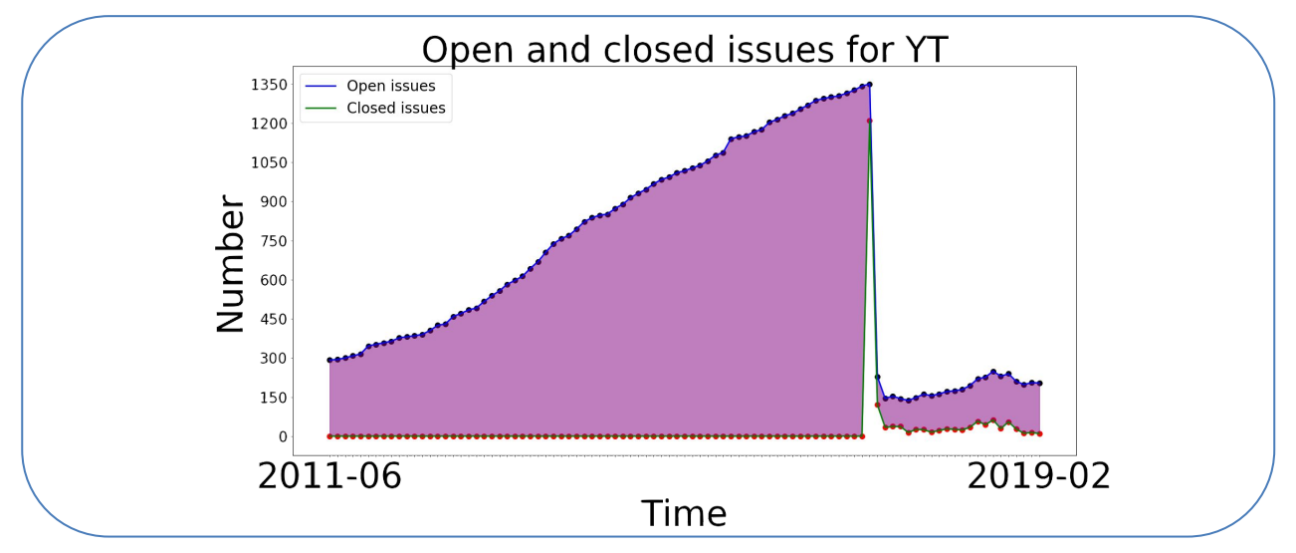
How engaged is the user and developer community?

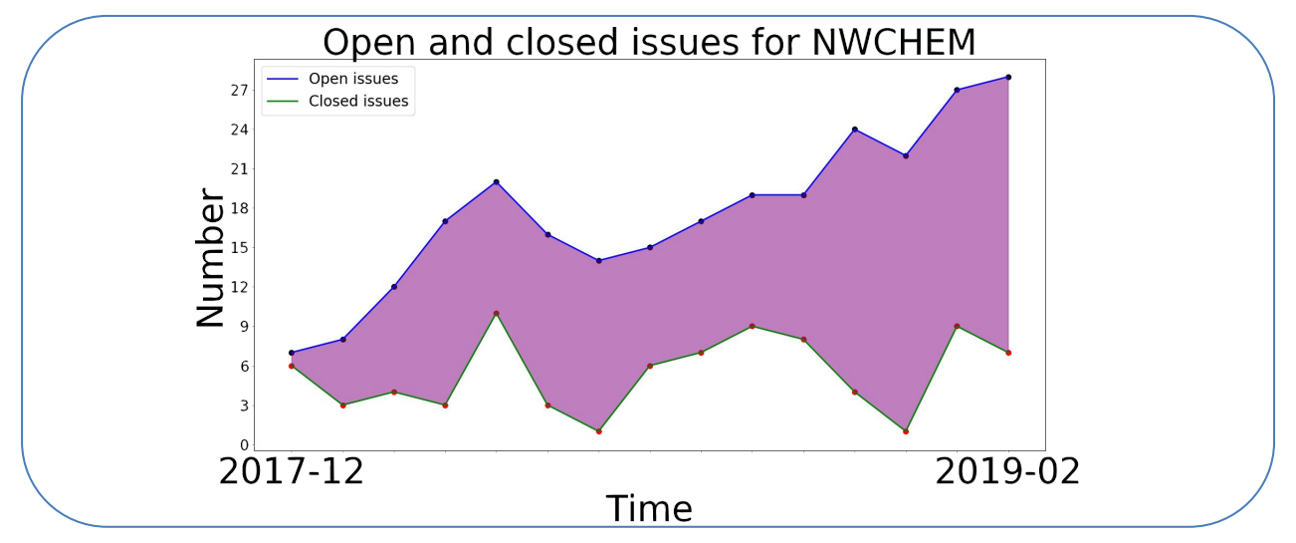


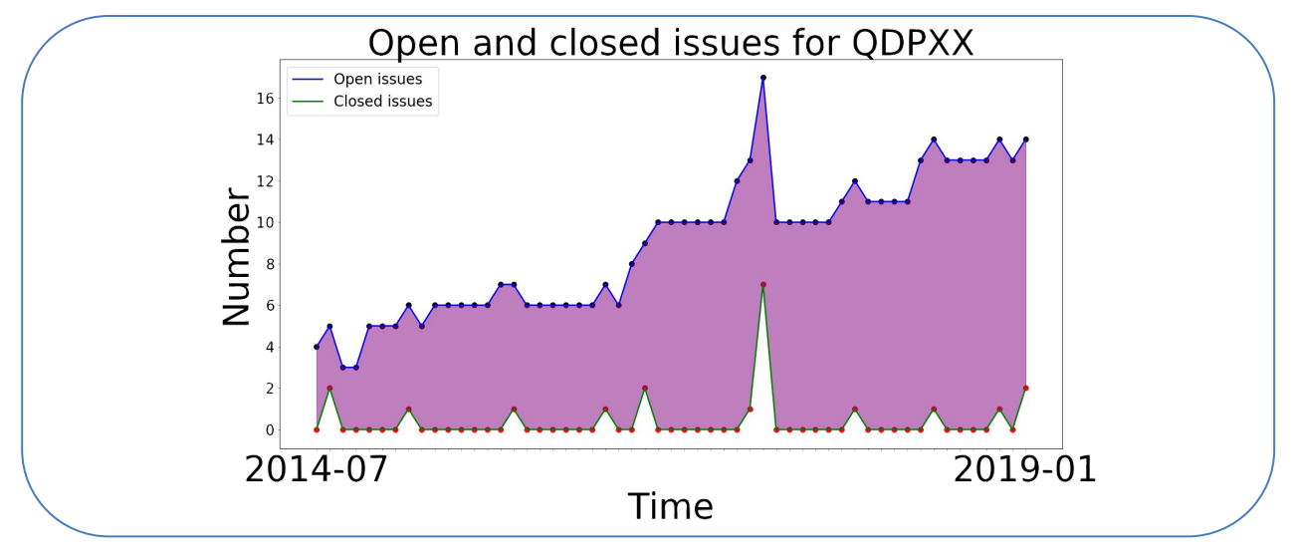












GROMACS: N/A: Uses Redmine for issue tracking which is not explored for now.

What topics dominate changes and discussion?

QMCPACK

Topic	No. of files changed
Rc 350	1971
Rc 360	1403

Topic	No. of comments
Optimizable determinants	91
MultiSlater-Jastrow Orbital Optimization code	84

SPACK

Topic	No. of files changed
Copyright 2019	3527
Relicense to Apache2/MIT	3386

Topic	No. of comments
Intel prefixes	266
do_install: allow for a no. of phases	201

MOOSE

Topic	No. of files changed
Create a separate test library for each module/app	5037
New Copyright proposal	4298

Topic	No. of comments
A semi-working push of our Threading feature	206
Cohesive Zone Modelling Via Interface Kernel	156

LAMMPS

Topic	No. of files changed
Restructure LAMMPS documentation folder	2523
Unstable	1633

Topic	No. of comments
Permanent PLUMED interface	84
cmake: build libs separate	50

YT

Topic	No. of files changed
Remove analysis_modules	194
Merging yt-4.0 with master	142

Topic	No. of comments
yt-4.0 adding "gather" approach	101
yt-4.0 adding octree support	92

NWCHEM

Topic	No. of files changed
Add support for Python 3	24
Remove DEC OS/F support	21

Topic	No. of comments
libnwchem patch	8
Add support for Python 3	6

QDPXX

Topic	No. of files changed
Move all include statements outside of namespaces	55
HDF5	43

Topic	No. of comments
fixed missing subtype.field() bug	3
in LatticePropagatorF3 writer, changed an Nc that should be Ns	2

GROMACS

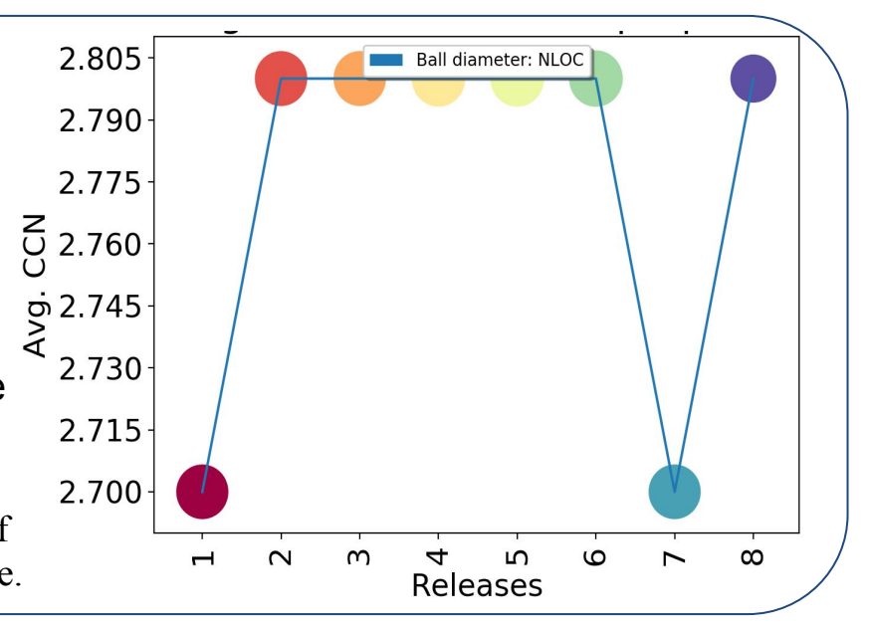
Topic	No. of files changed
Virial	225

Topic	No. of comments
Update gurgle.dat	3

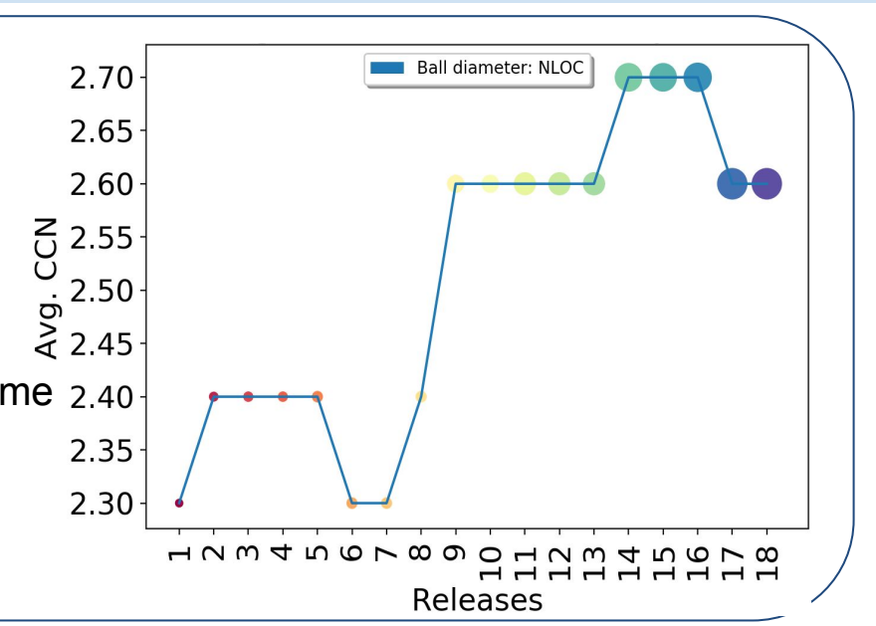
How is code complexity changing over time?

- **CCN***: Cyclomatic complexity
- **Releases**: Project releases over time
- **Color**: Change in time

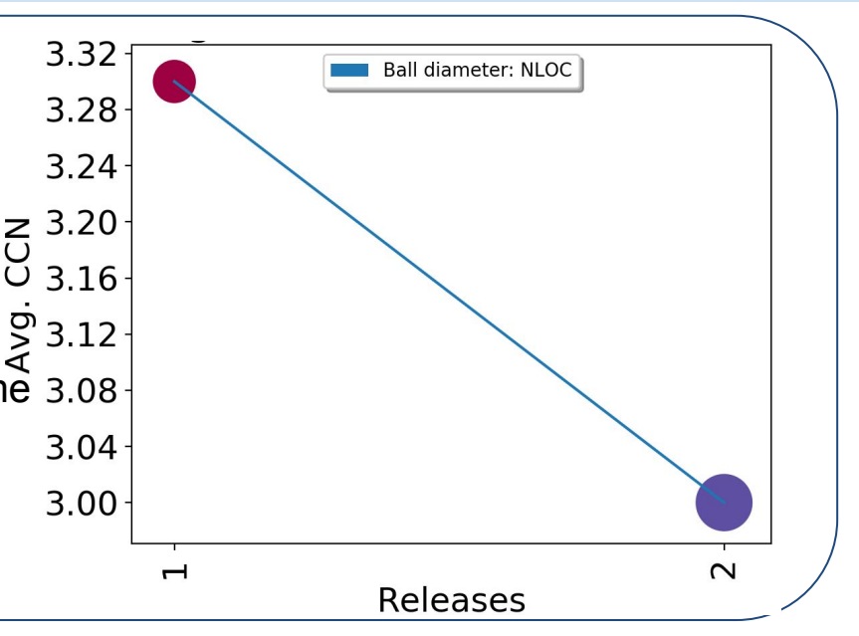
CCN*: Quantitative measure of the number of independent paths in code.



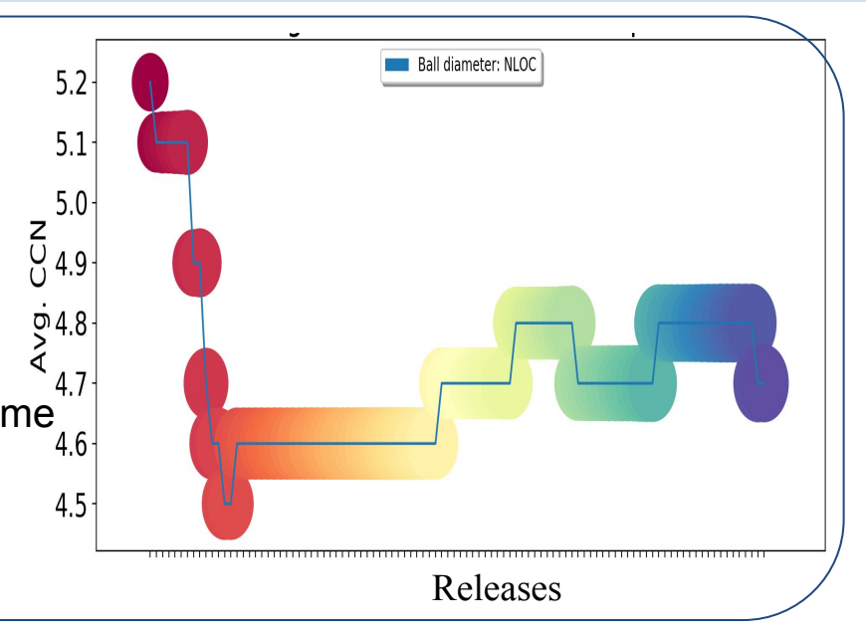
- **CCN***: Cyclomatic complexity
- **Releases**: Project releases over time
- **Color**: Change in time



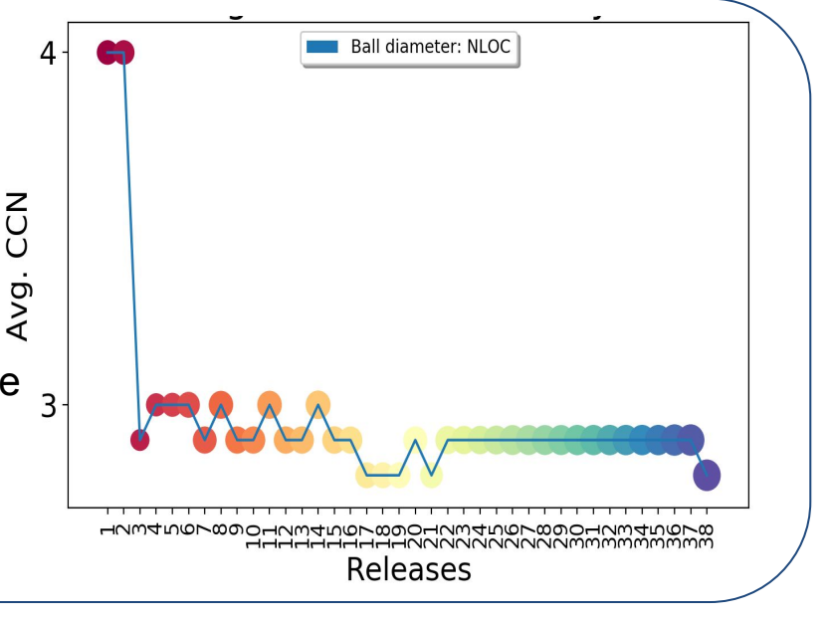
- **CCN***: Cyclomatic complexity
- **Releases**: Project releases over time
- **Color**: Change in time



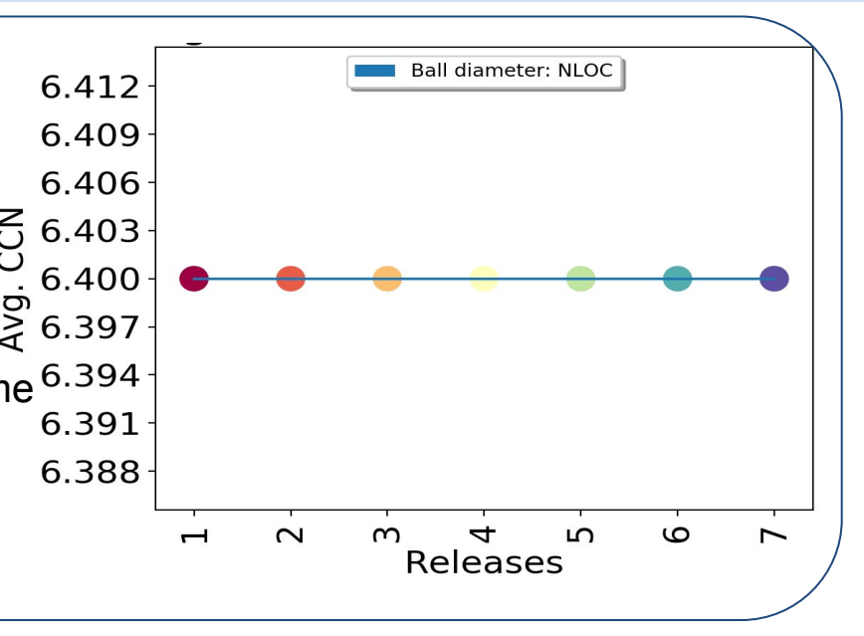
- **CCN***: Cyclomatic complexity
- **Releases**: Project releases over time
- **Color**: Change in time



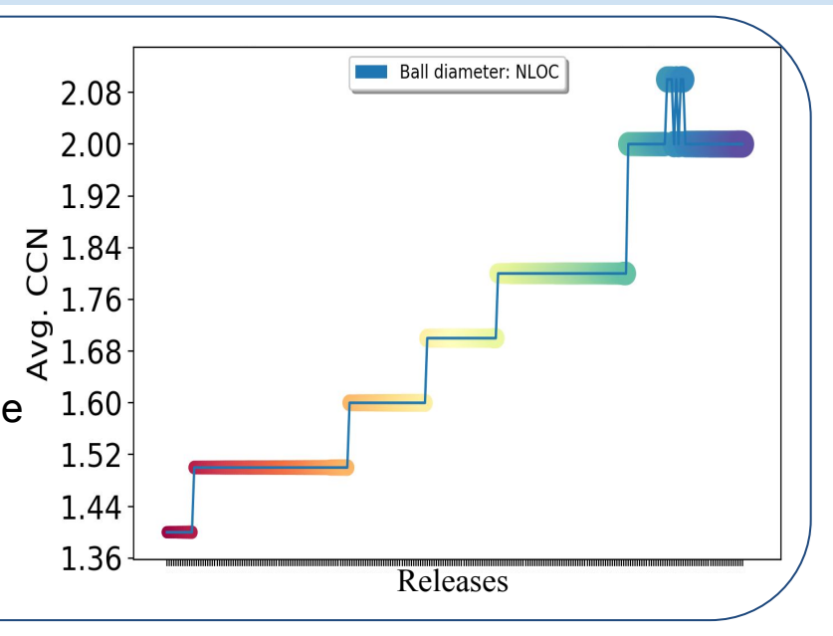
- **CCN***: Cyclomatic complexity
- **Releases**: Project releases over time
- **Color**: Change in time



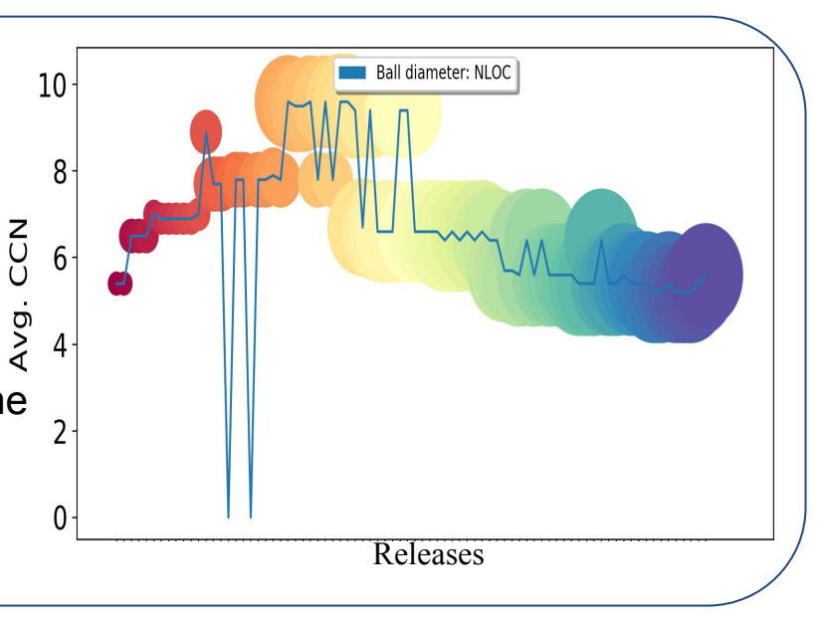
- **CCN***: Cyclomatic complexity
- **Releases**: Project releases over time
- **Color**: Change in time



- **CCN***: Cyclomatic complexity
- **Releases**: Project releases over time
- **Color**: Change in time



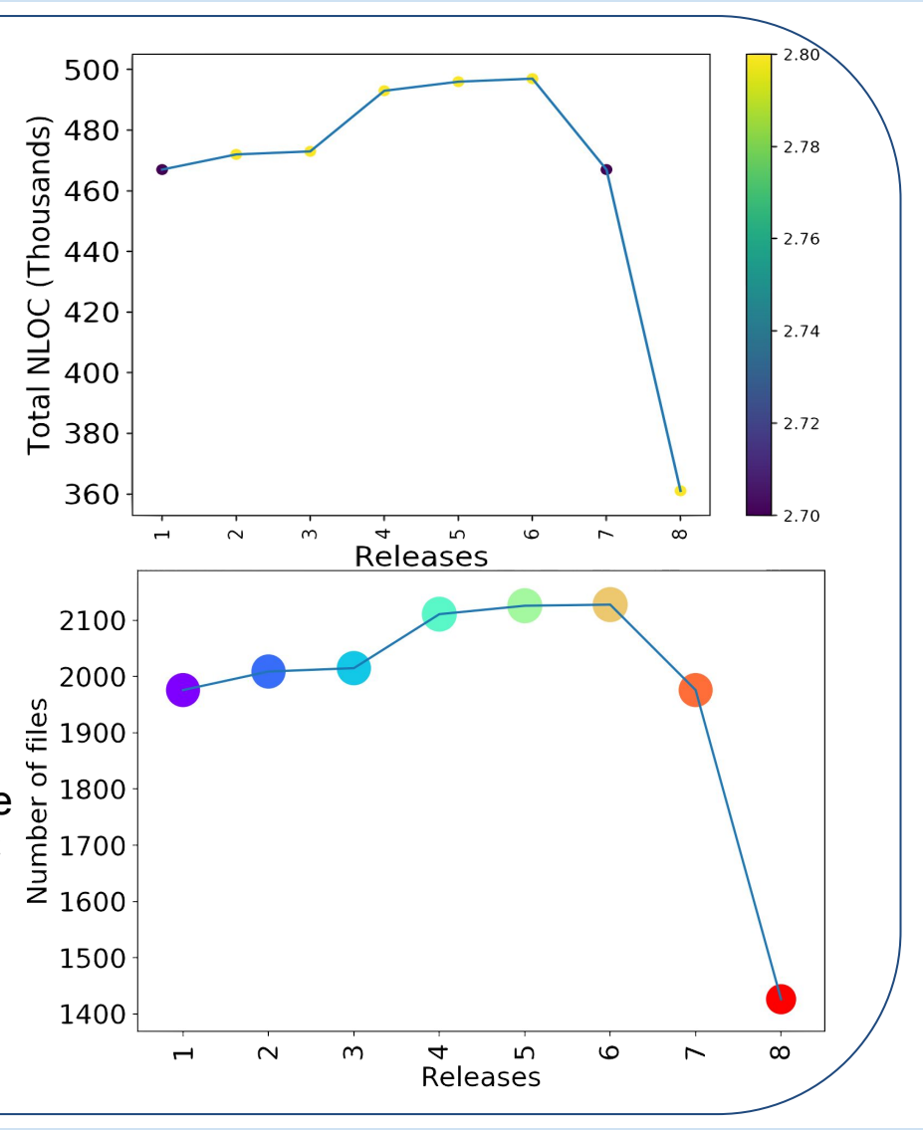
- **CCN***: Cyclomatic complexity
- **Releases**: Project releases over time
- **Color**: Change in time



How is the code size changing over time?

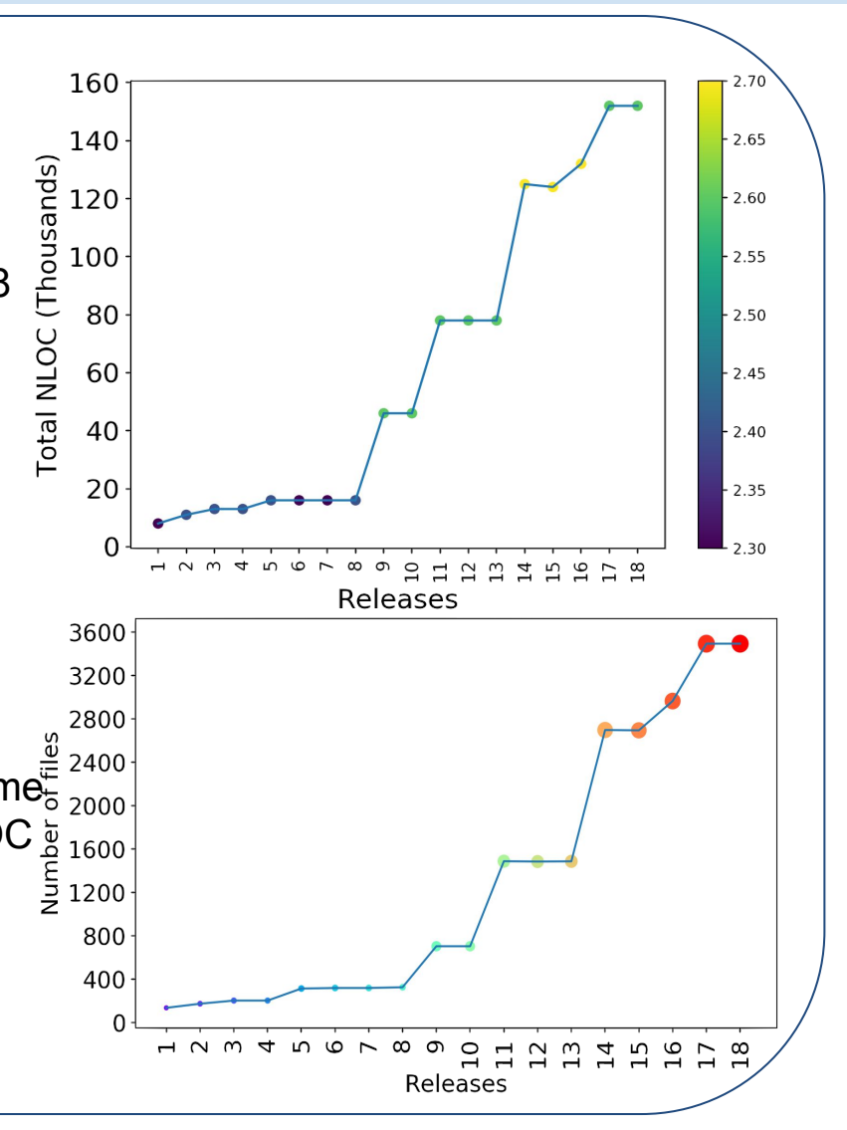
- **Releases**: Project releases over time
- **No. of releases**: 8
- **Color**: Avg. CCN

- **Files**: Code files
- **Releases**: Project releases over time
- **Color**: Change in time
- **Ball diameter**: NLOC



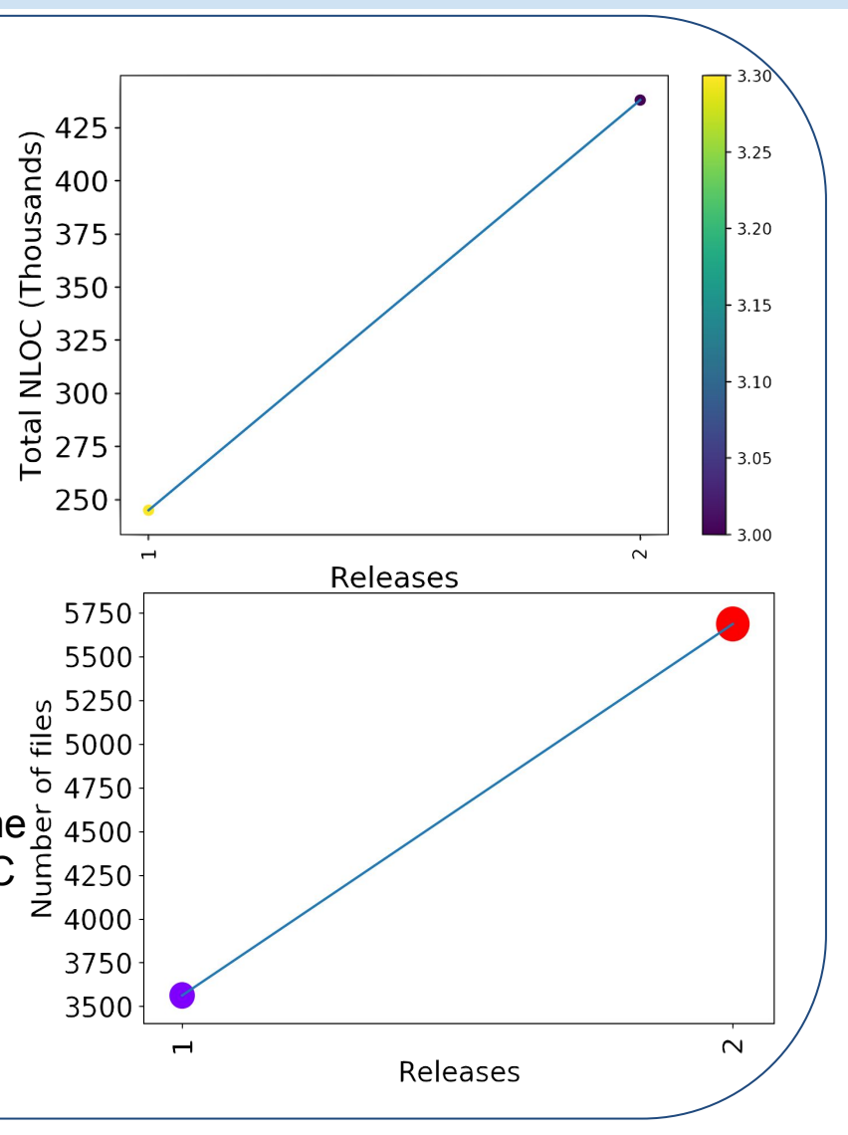
- **Releases**: Project releases over time
- **No. of releases**: 18
- **Color**: Avg. CCN

- **Files**: Code files
- **Releases**: Project releases over time
- **Color**: Change in time
- **Ball diameter**: NLOC



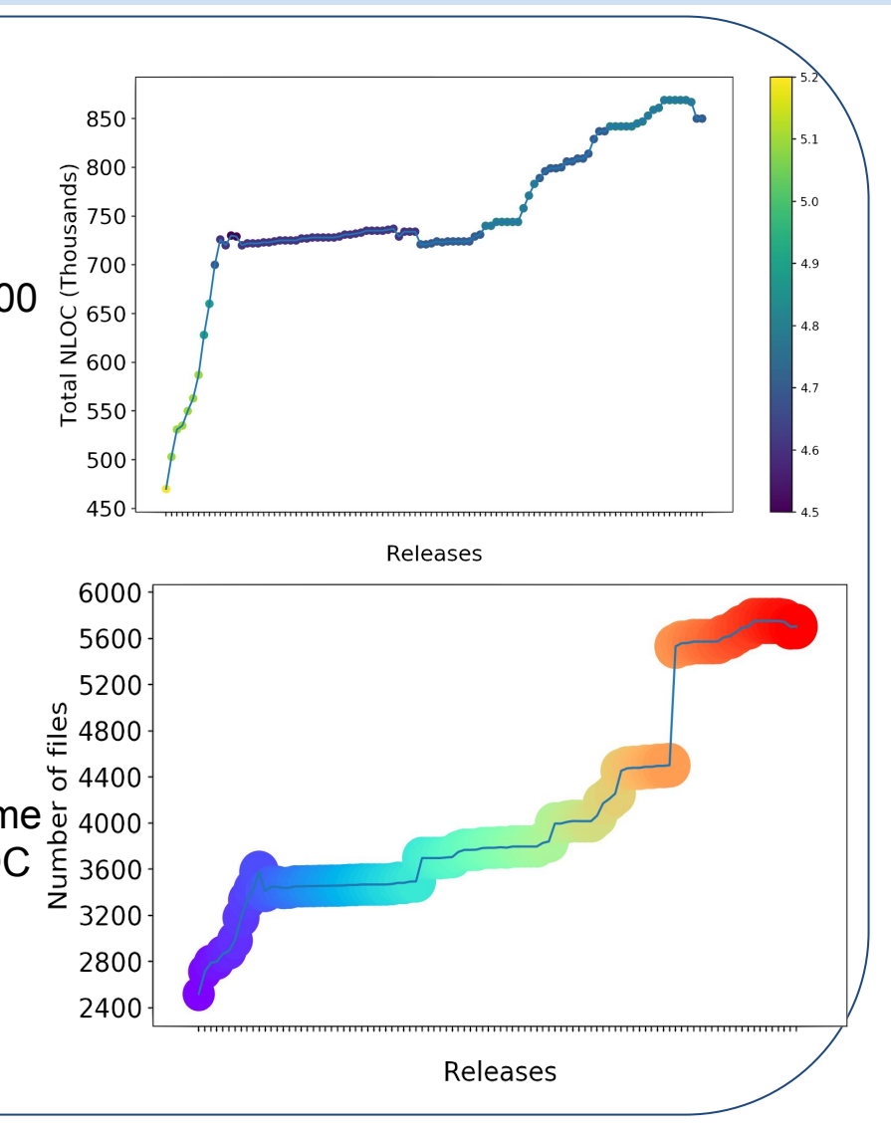
- **Releases**: Project releases over time
- **No. of releases**: 2
- **Color**: Avg. CCN

- **Files**: Code files
- **Releases**: Project releases over time
- **Color**: Change in time
- **Ball diameter**: NLOC



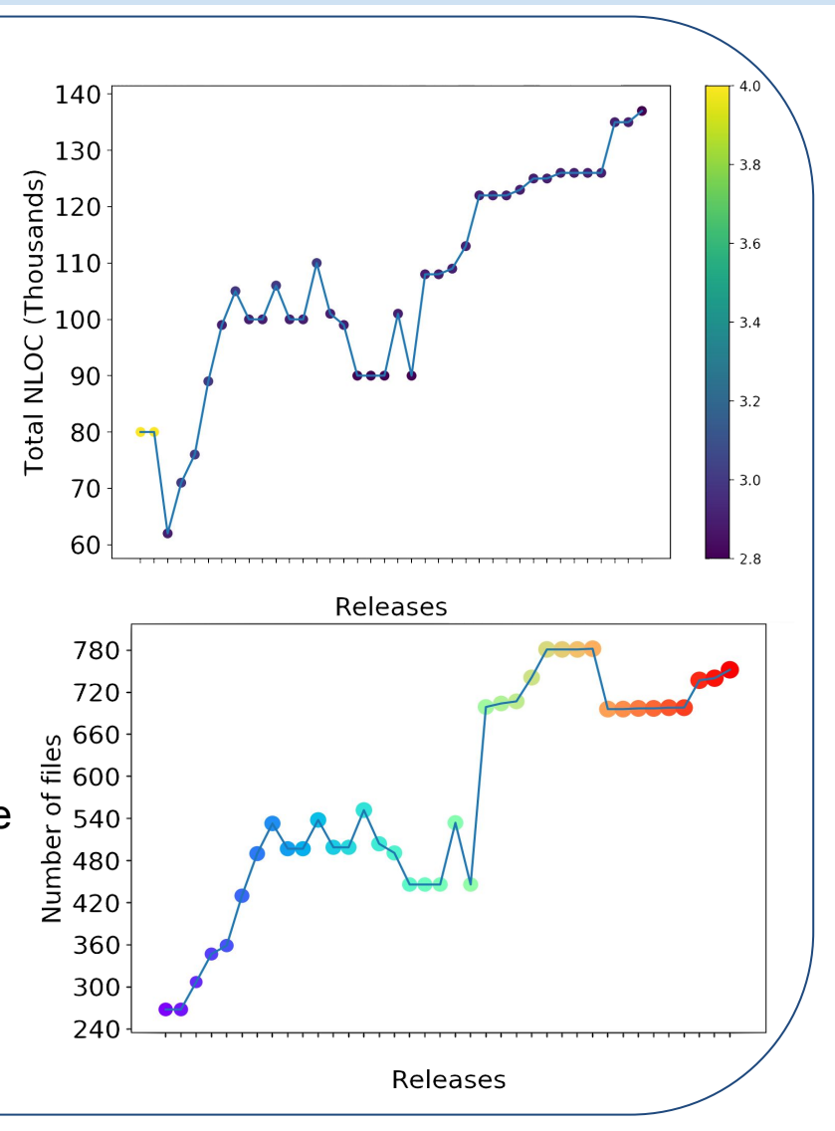
- **Releases**: Project releases over time
- **No. of releases**: 100
- **Color**: Avg. CCN

- **Files**: Code files
- **Releases**: Project releases over time
- **Color**: Change in time
- **Ball diameter**: NLOC



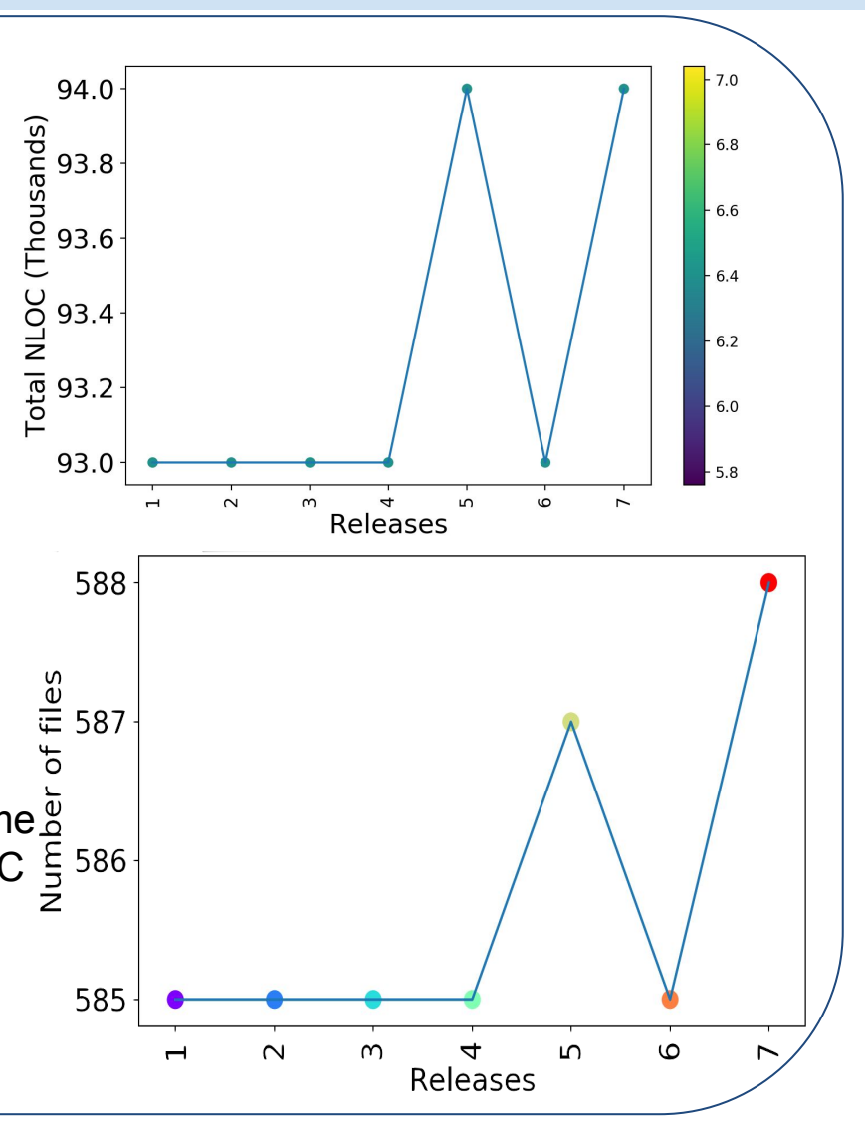
- **Releases**: Project releases over time
- **No. of releases:** 38
- **Color**: Avg. CCN

- **Files**: Code files
- **Releases**: Project releases over time
- **Color**: Change in time
- **Ball diameter**: NLOC



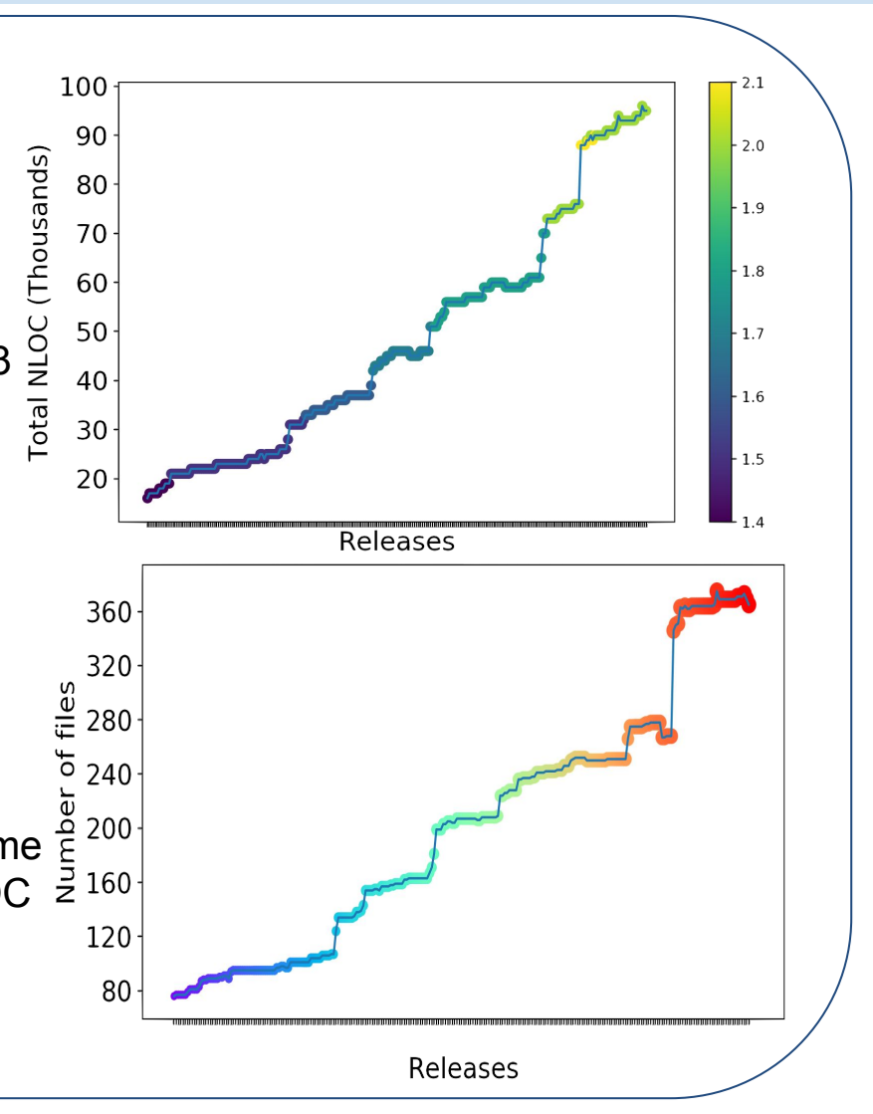
- **Releases**: Project releases over time
- **No. of releases**: 7
- **Color**: Avg. CCN

- **Files**: Code files
- **Releases**: Project releases over time
- **Color**: Change in time
- **Ball diameter**: NLOC



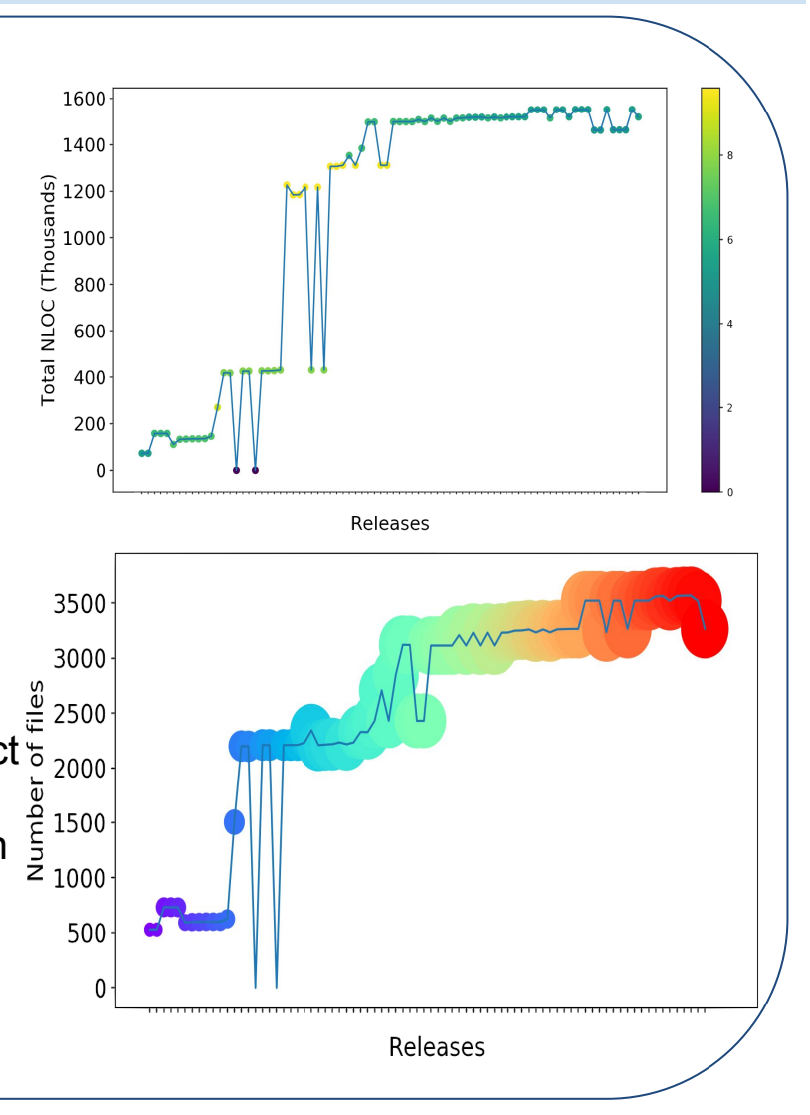
- **Releases**: Project releases over time
- **No. of releases**: 253
- **Color**: Avg. CCN

- **Files**: Code files
- **Releases**: Project releases over time
- **Color**: Change in time
- **Ball diameter**: NLOC

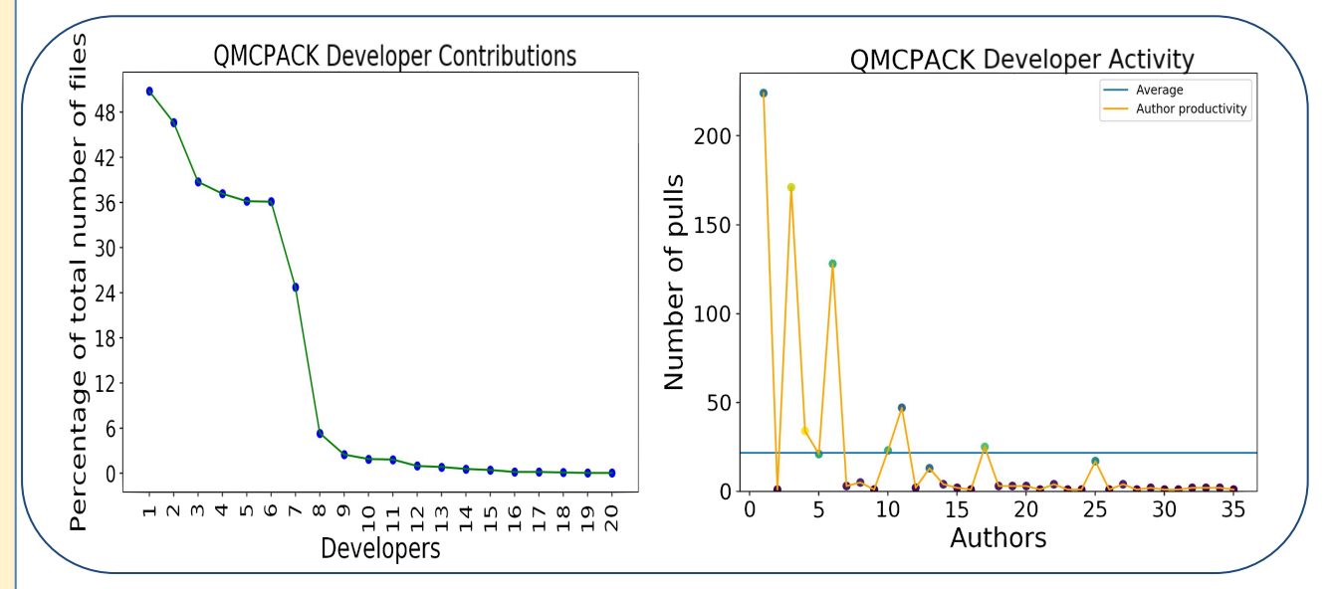


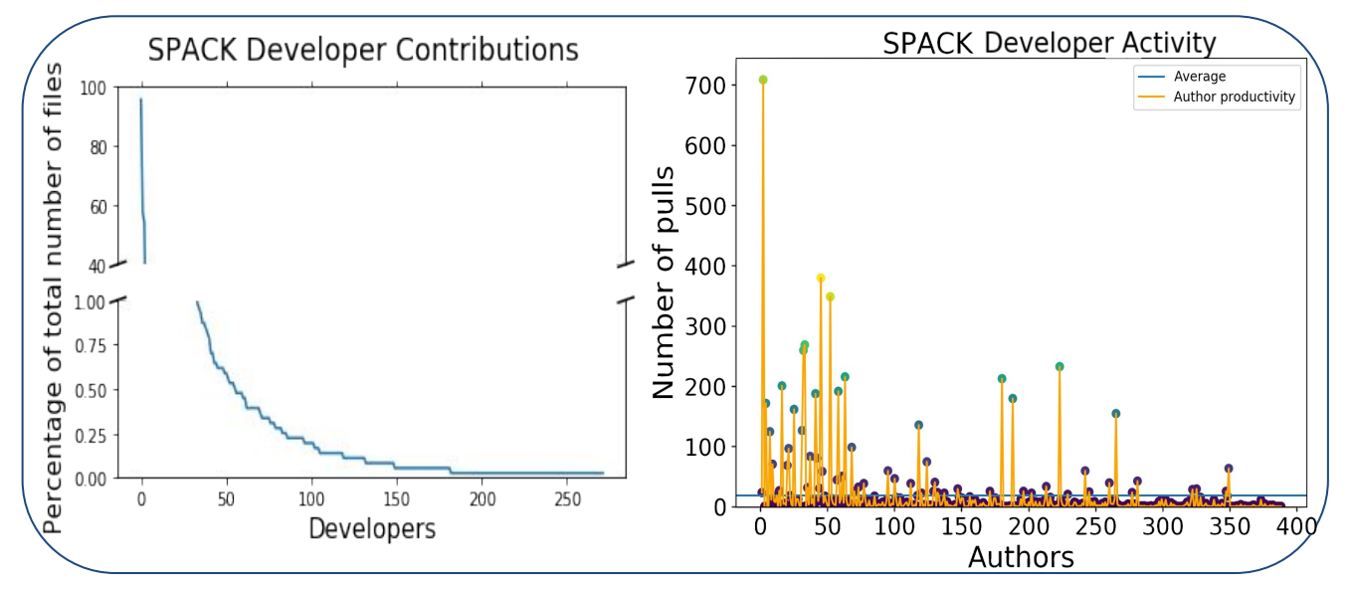
- **Releases**: Project releases over time
- **No. of releases**: 80
- **Color**: Avg. CCN

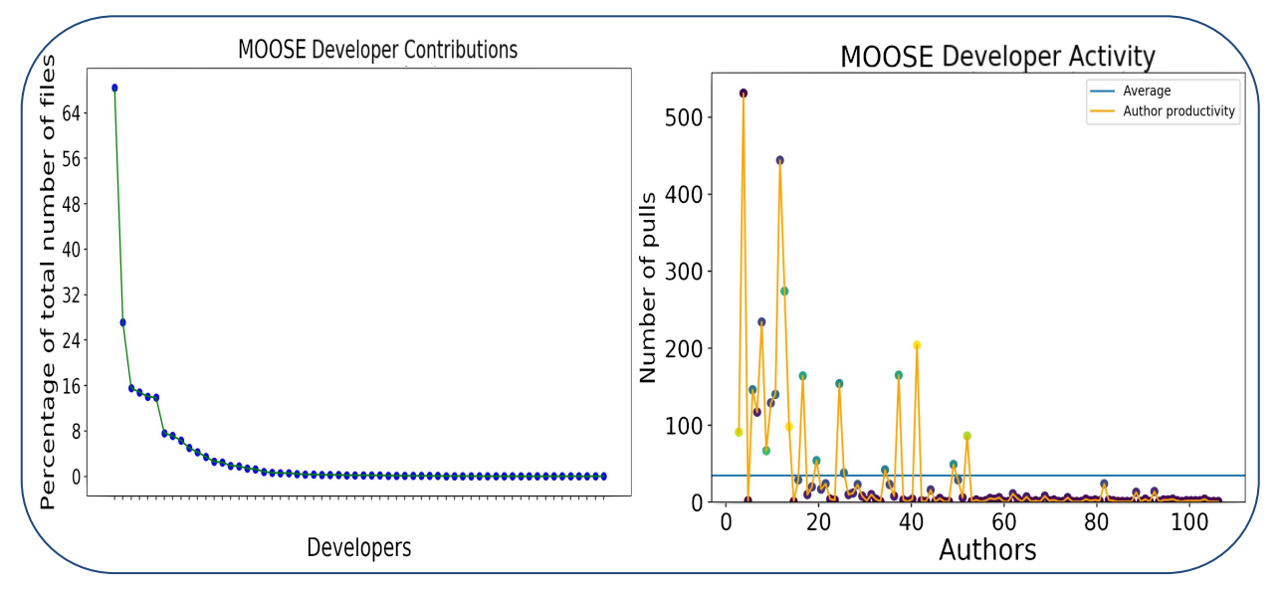
- **Files**: Code files
- **Releases**: Project releases over time
- **Color**: Change in time
- **Ball diameter**: NLOC

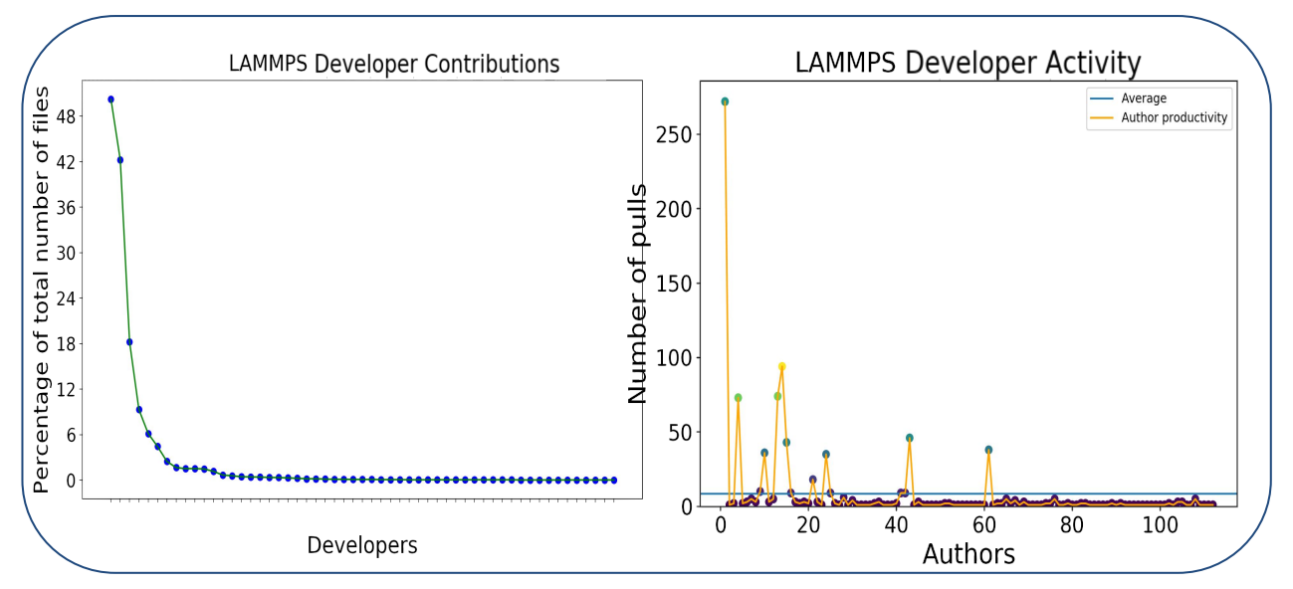


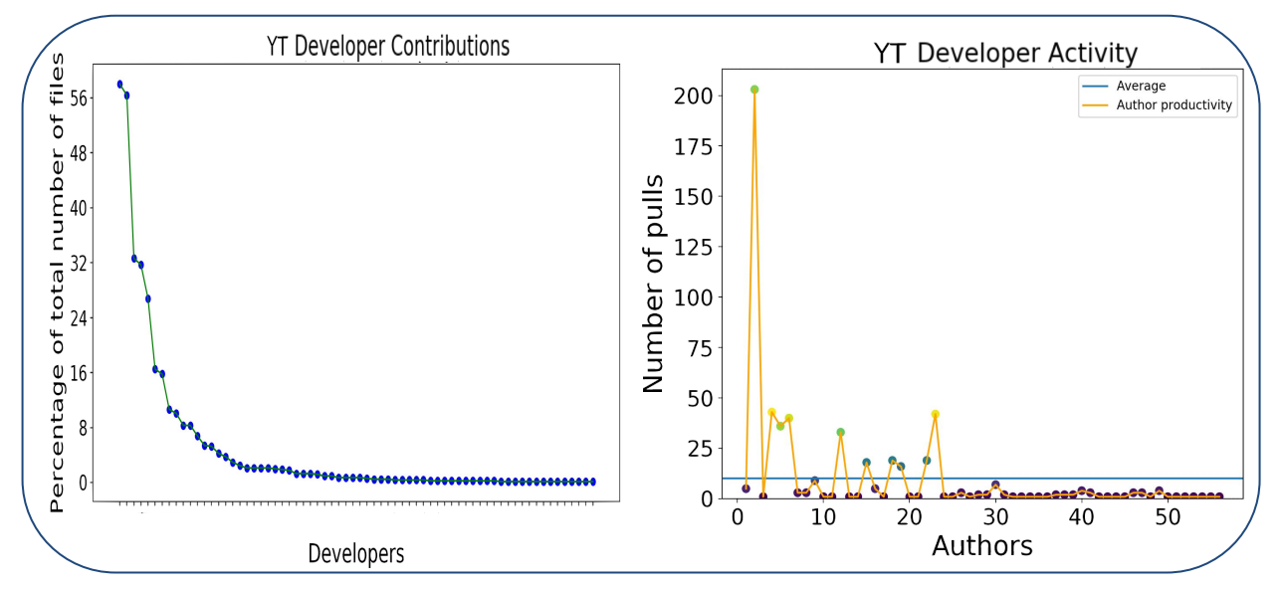
How active is the developer community?

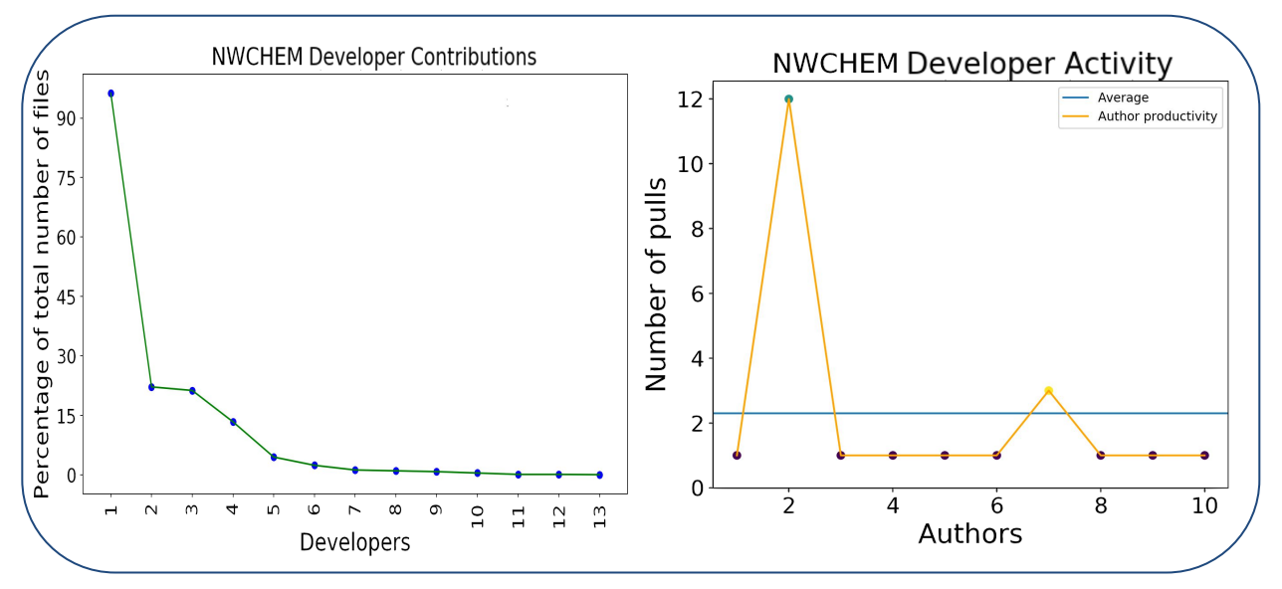


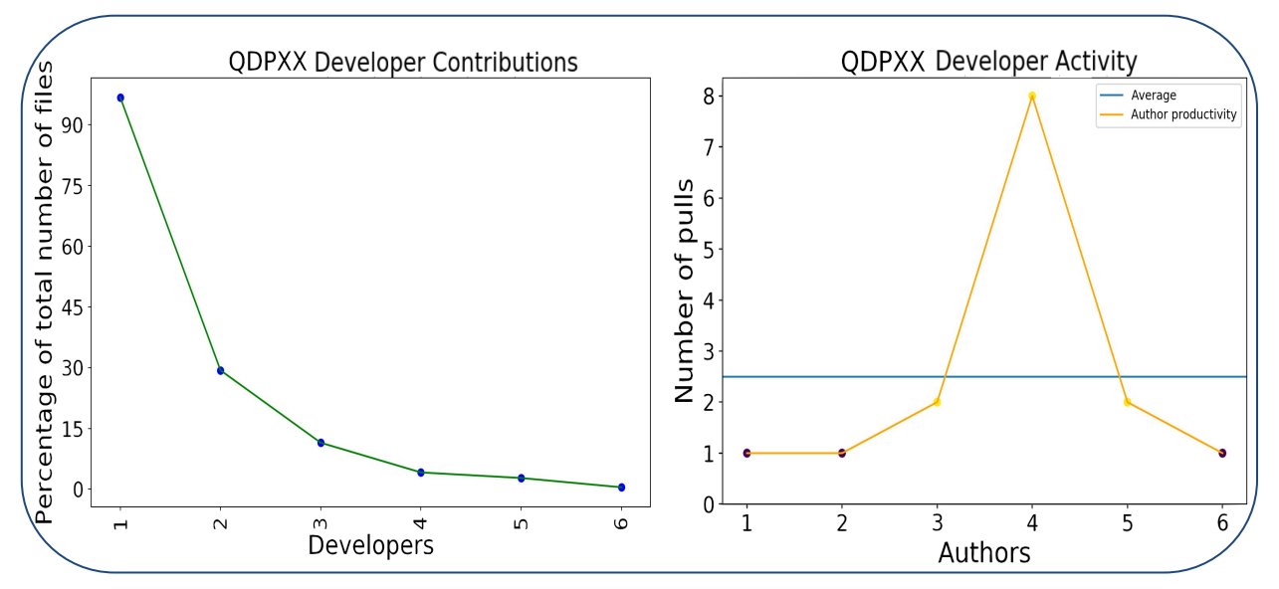


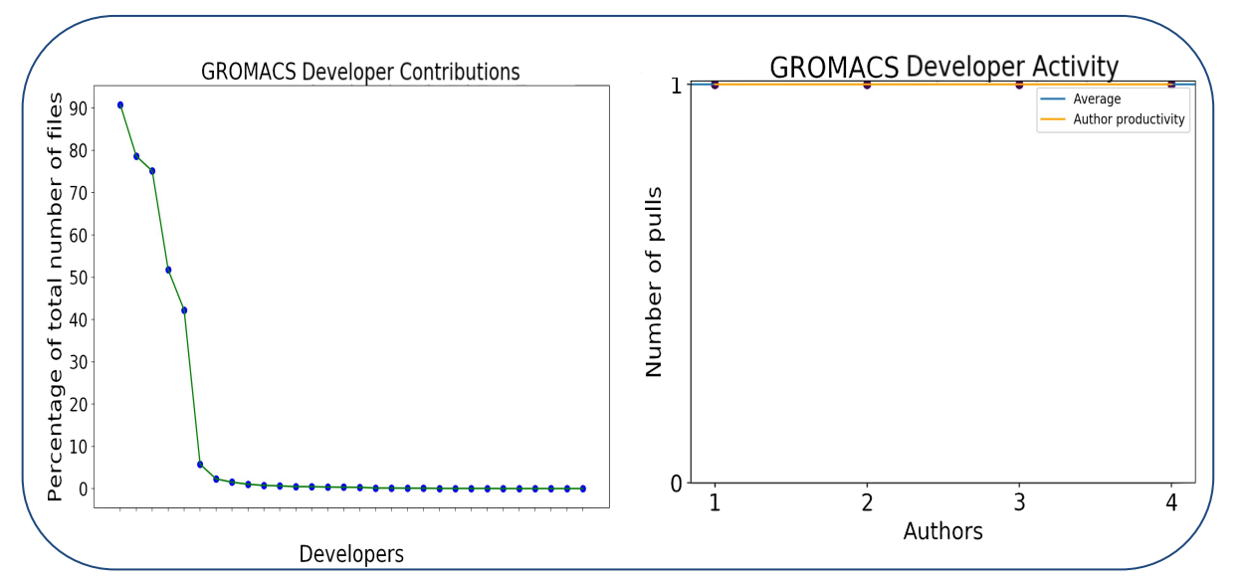












Disclaimer: The goal of this research is to explore new software metrics that can provide insight into productivity more effectively than existing metrics. These (or any) metrics provide partial perspectives but cannot capture a complete view of the complexities of scientific software projects.