UNIVERSITAT JAUME I

Towards Continuous Benchmarking:

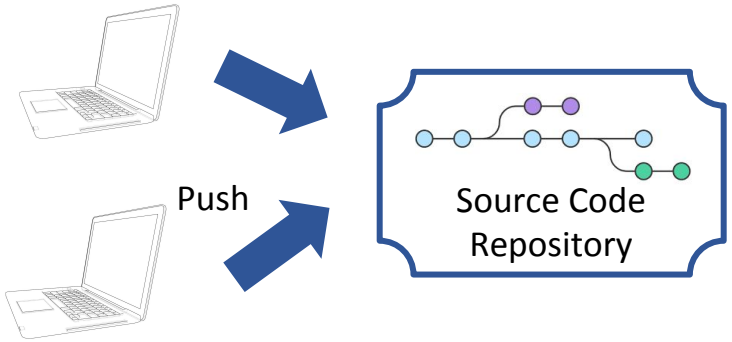
An Automated Performance Evaluation Framework for High Performance Software

Anzt, Chen, Cojean, Dongarra, Flegar, Nayak, Quintana-Ortí, Tsai, Wang

A Healthy Software Development Cycle

Developer

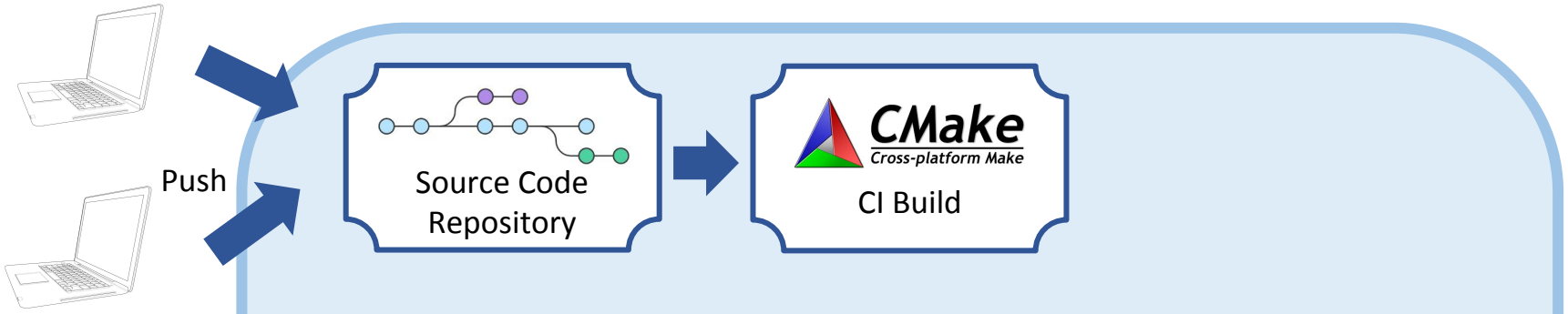
A Healthy Software Development Cycle



Developer

- Version control system for tracking changes and coordinating collaborative development.

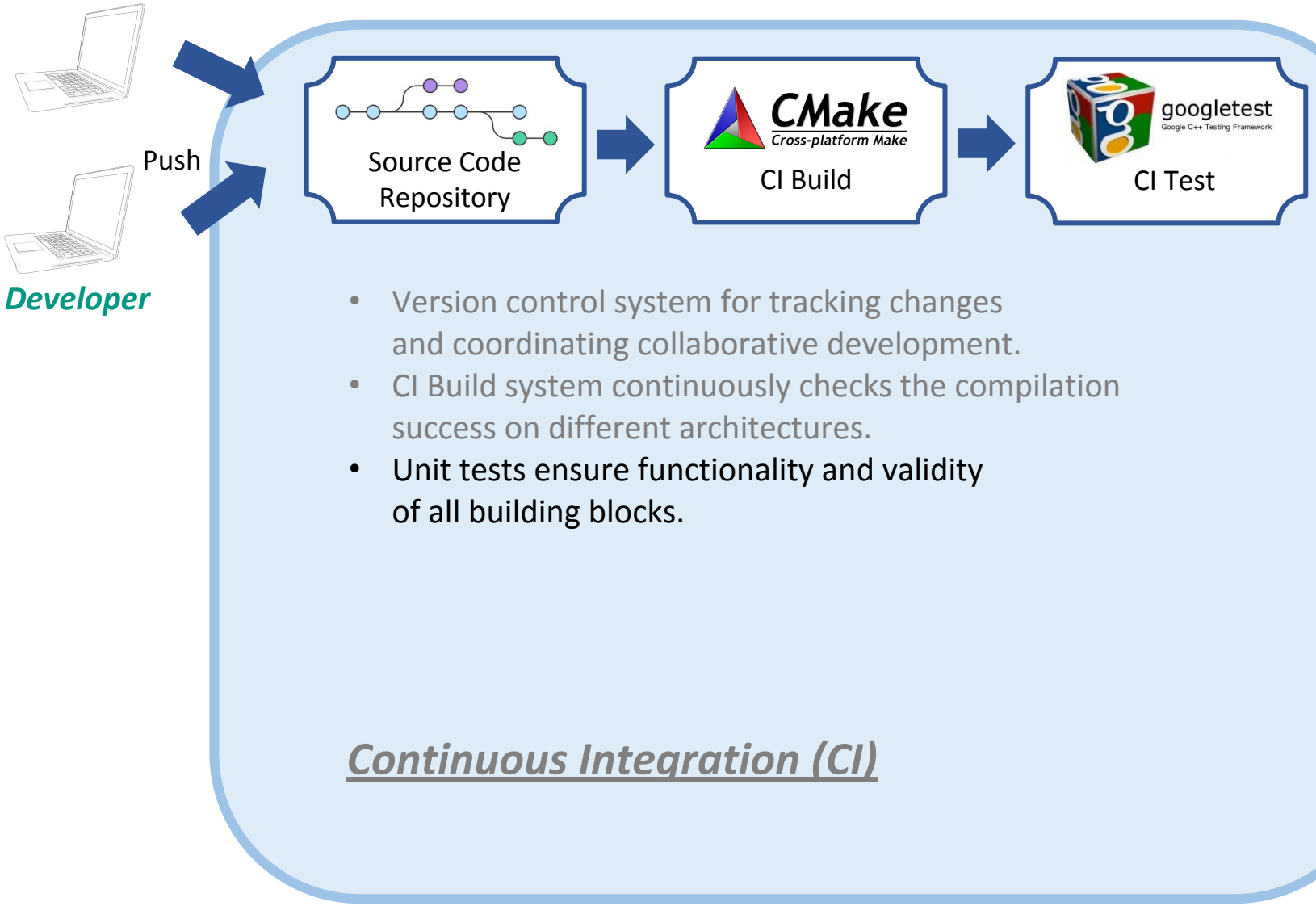
A Healthy Software Development Cycle



- Version control system for tracking changes and coordinating collaborative development.
- CI Build system continuously checks the compilation success on different architectures.

Continuous Integration (CI)

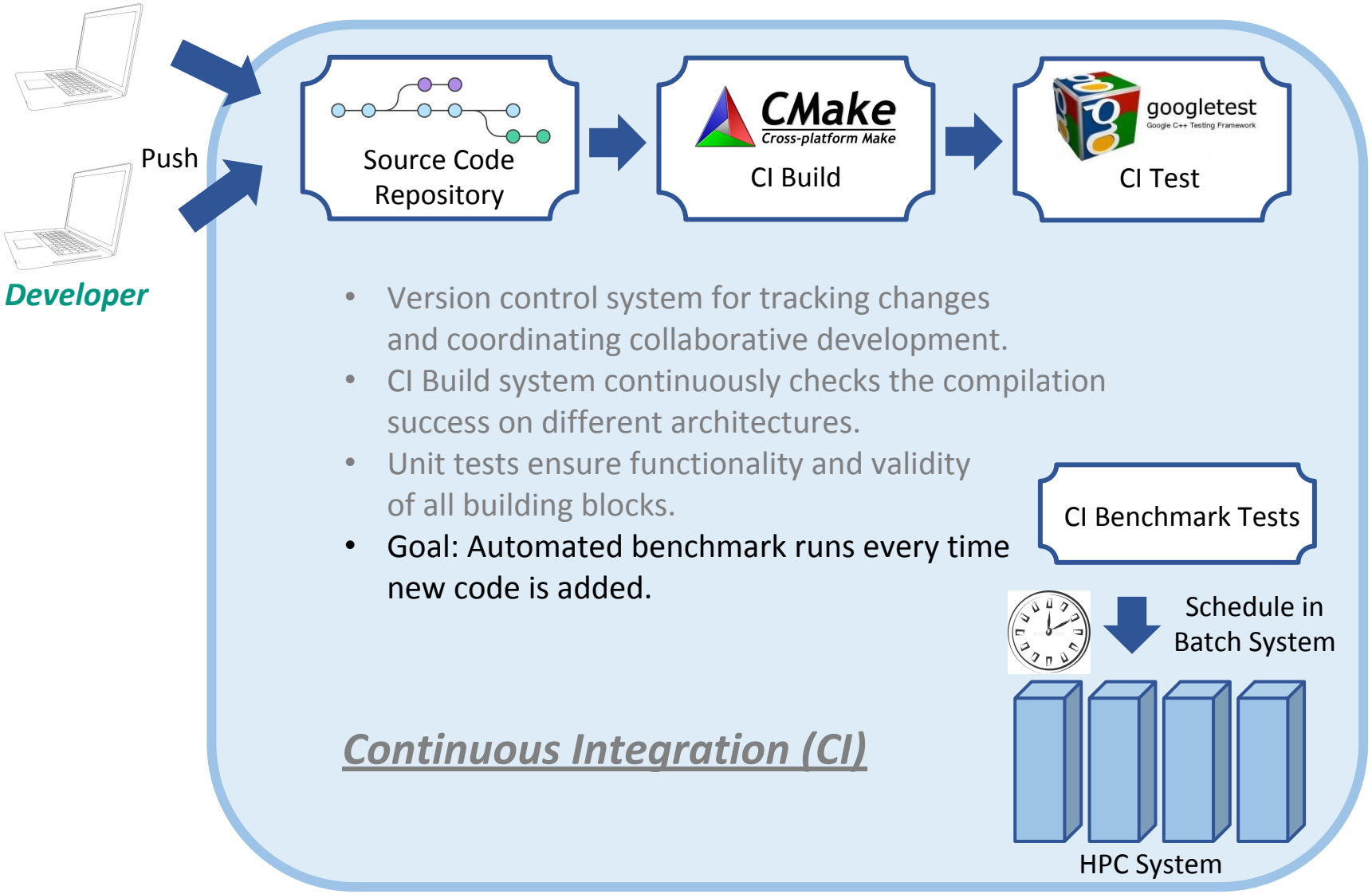
A Healthy Software Development Cycle



- Version control system for tracking changes and coordinating collaborative development.
- CI Build system continuously checks the compilation success on different architectures.
- Unit tests ensure functionality and validity of all building blocks.

Continuous Integration (CI)

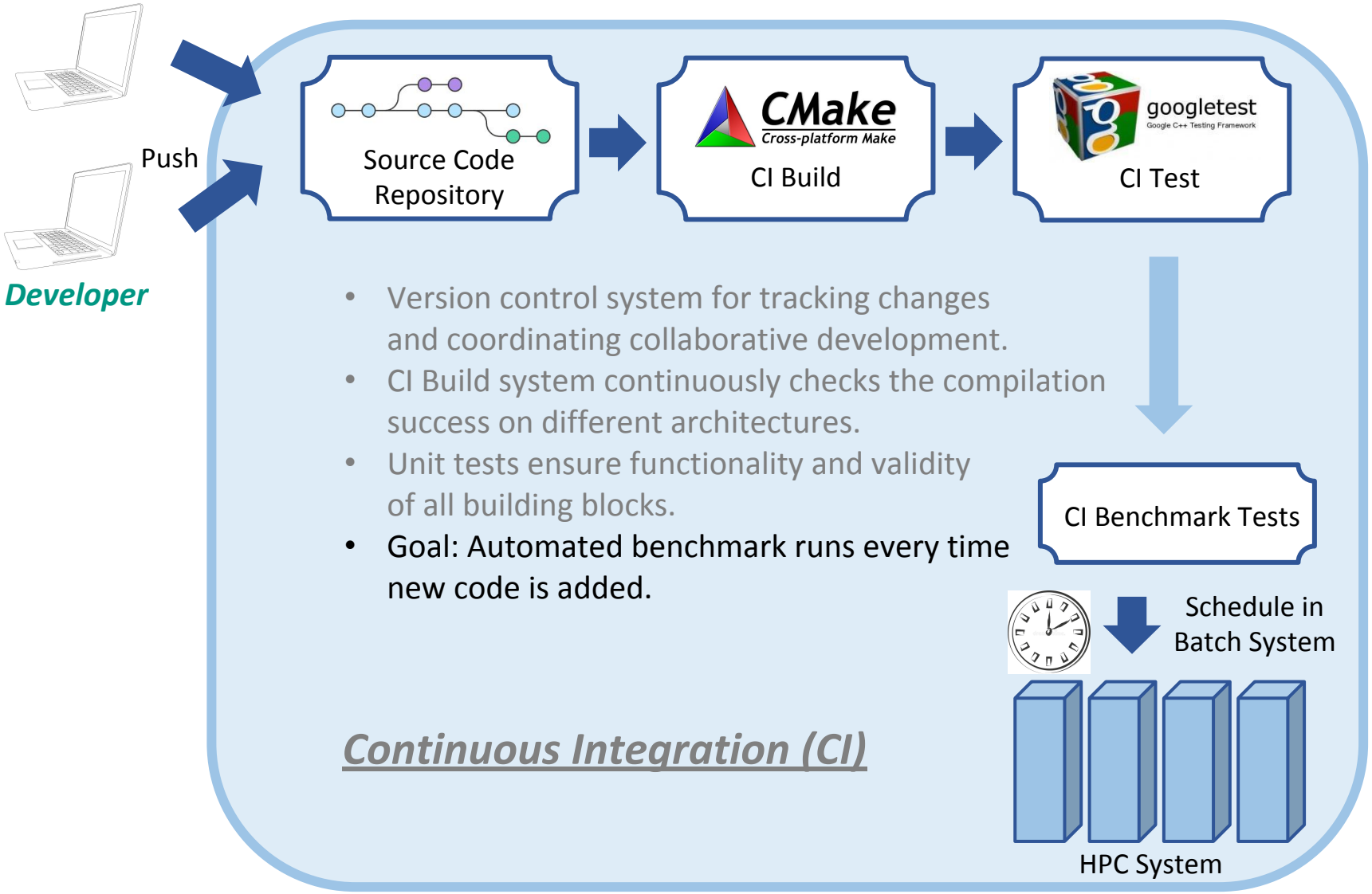
A Healthy Software Development Cycle



- Version control system for tracking changes and coordinating collaborative development.
- CI Build system continuously checks the compilation success on different architectures.
- Unit tests ensure functionality and validity of all building blocks.
- Goal: Automated benchmark runs every time new code is added.

Continuous Integration (CI)

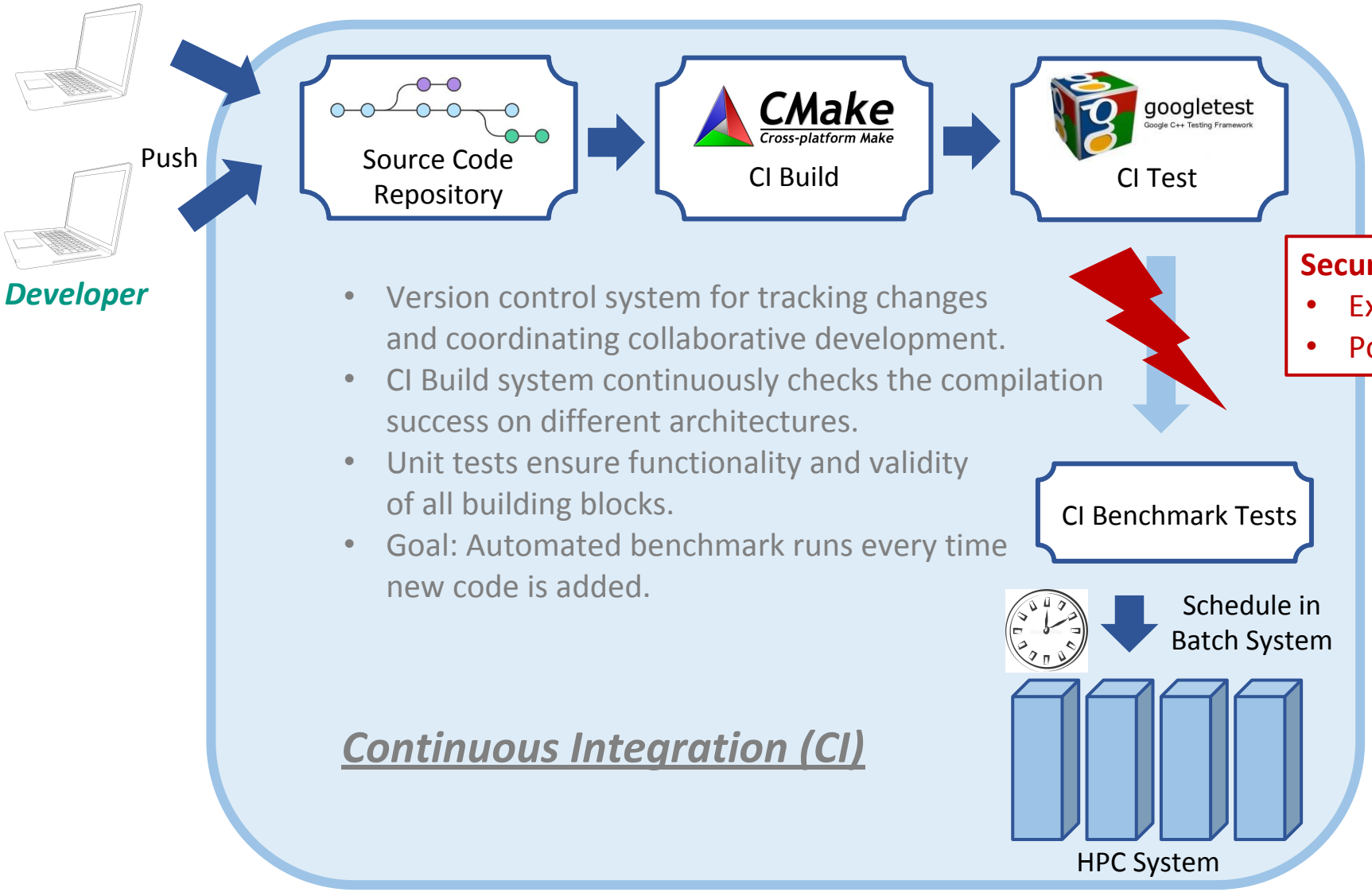
A Healthy Software Development Cycle



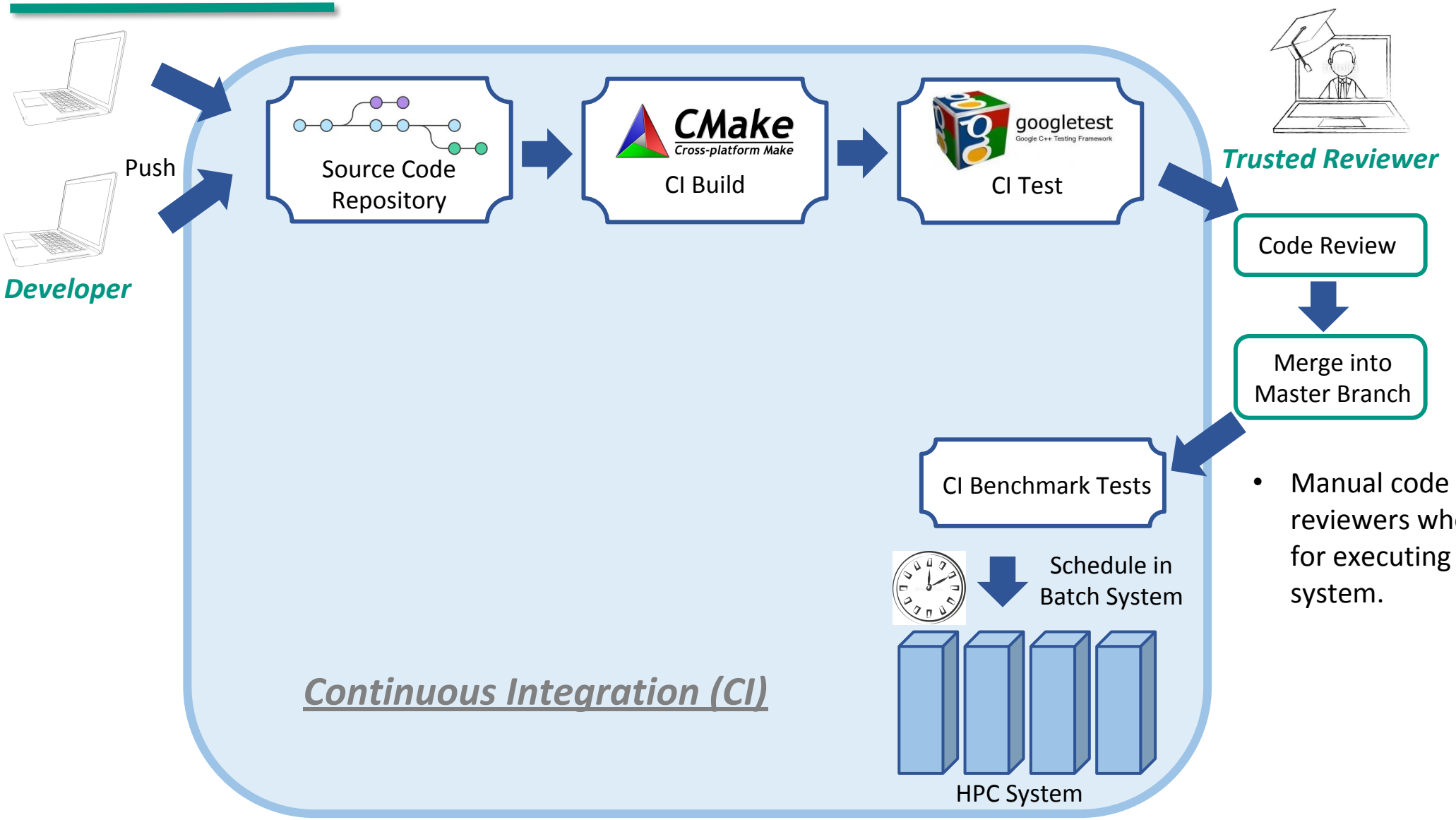
- Version control system for tracking changes and coordinating collaborative development.
- CI Build system continuously checks the compilation success on different architectures.
- Unit tests ensure functionality and validity of all building blocks.
- Goal: Automated benchmark runs every time new code is added.

Continuous Integration (CI)

A Healthy Software Development Cycle

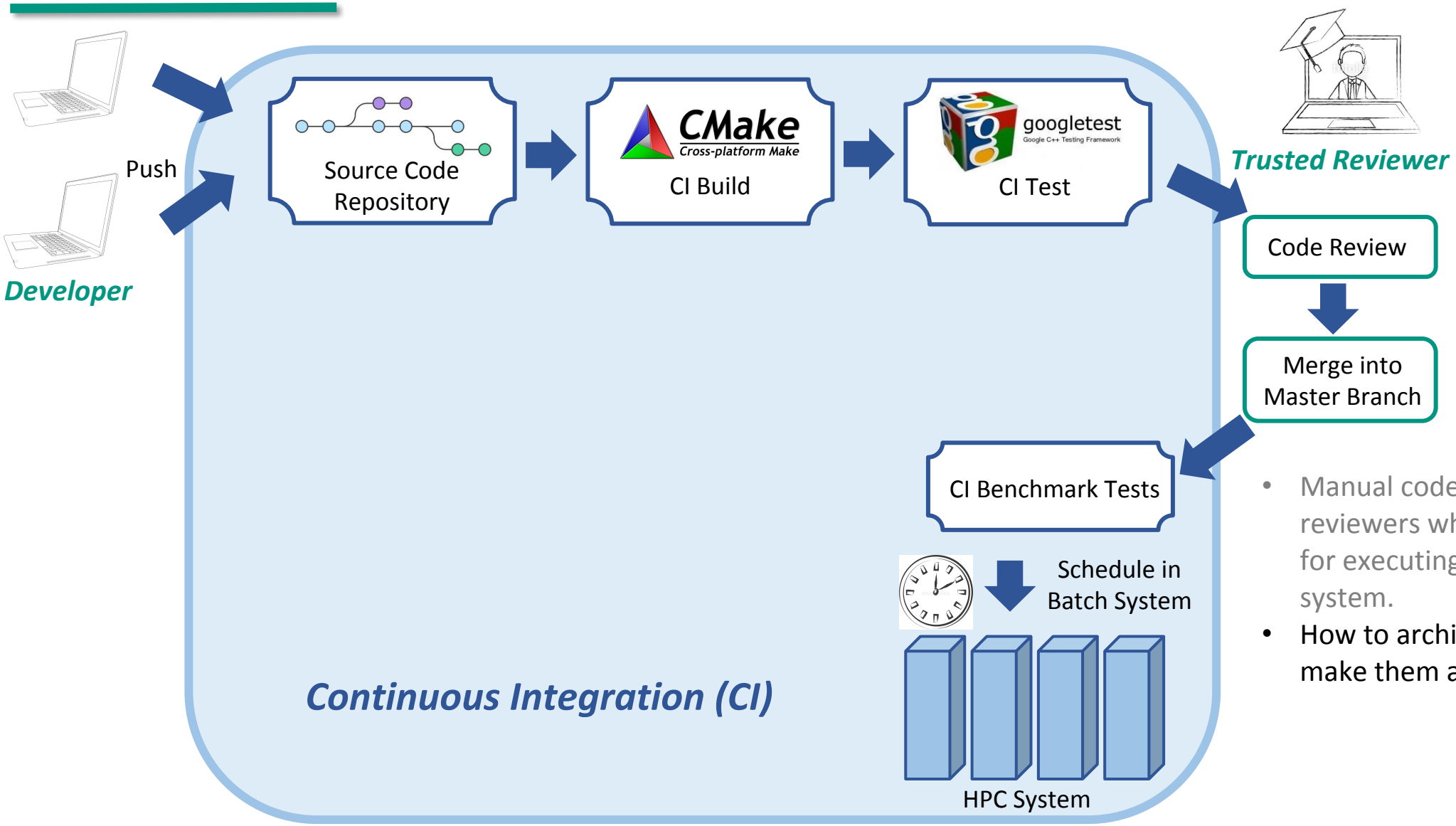


A Healthy Software Development Cycle



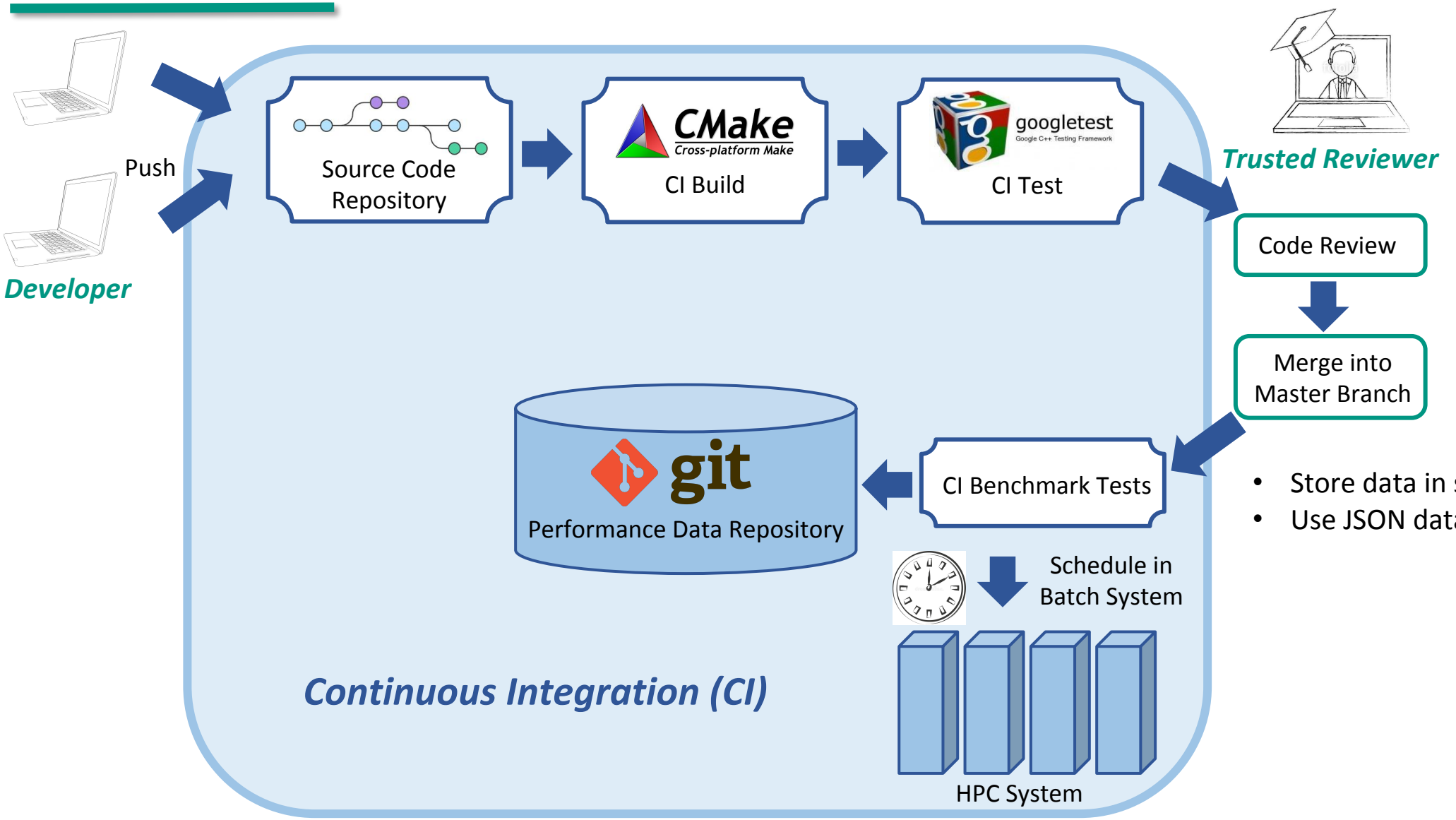
- Manual code reviews by trusted reviewers who take responsibility for executing code on the HPC system.

A Healthy Software Development Cycle



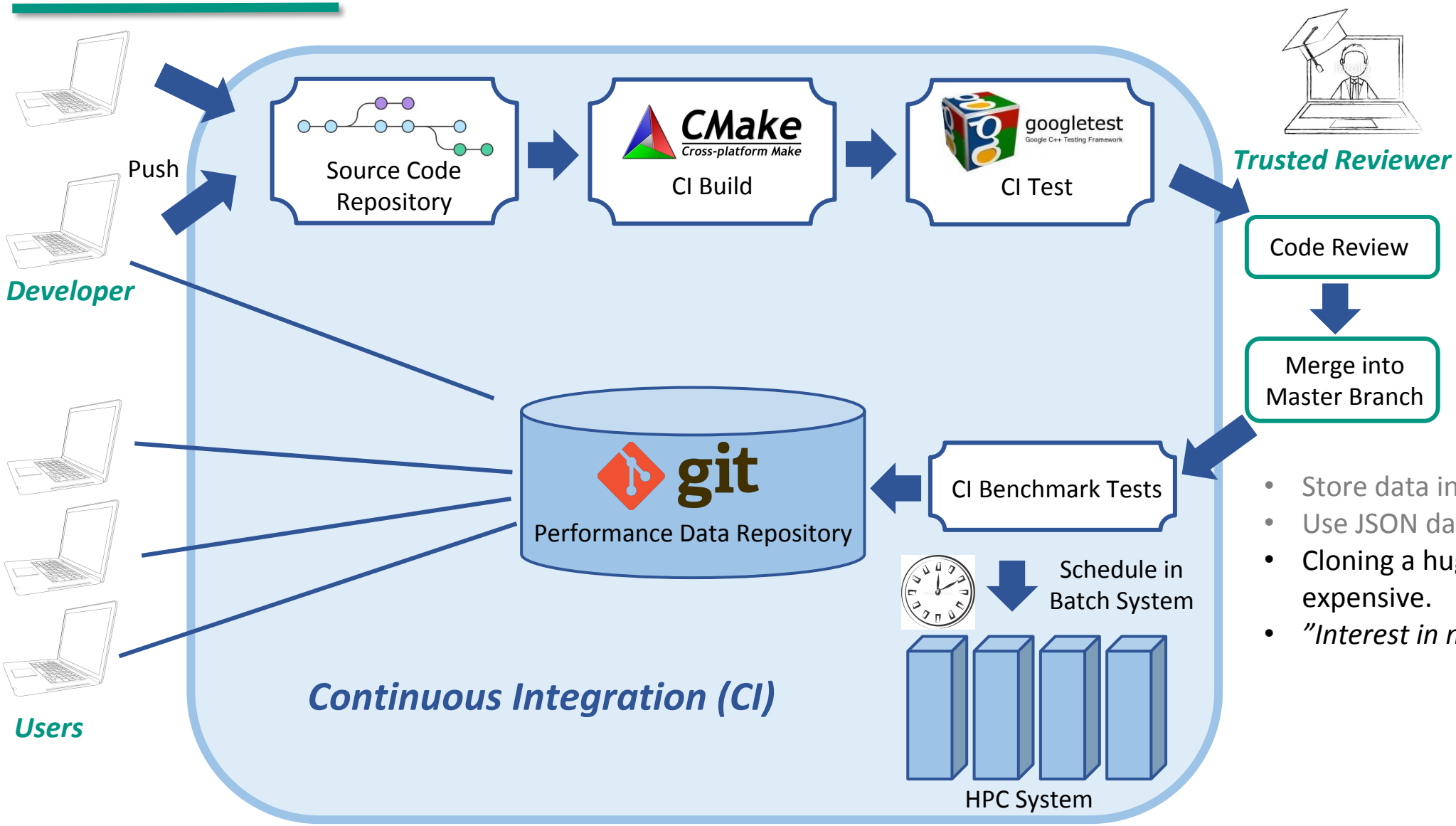
- Manual code reviews by trusted reviewers who take responsibility for executing code on the HPC system.
- How to archive the results and make them accessible?

A Healthy Software Development Cycle



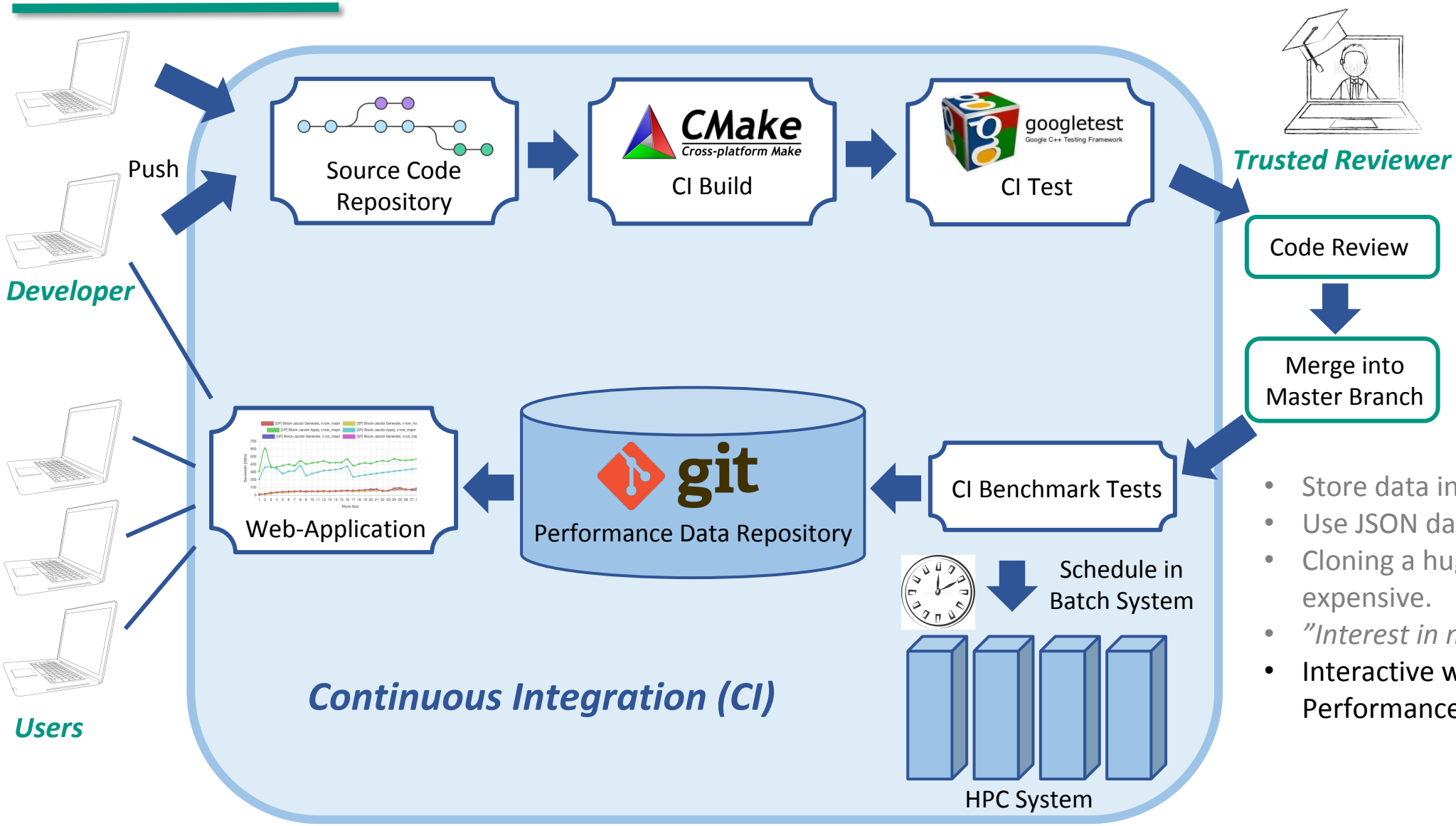
- Store data in separate repository.
- Use JSON data exchange format.

A Healthy Software Development Cycle



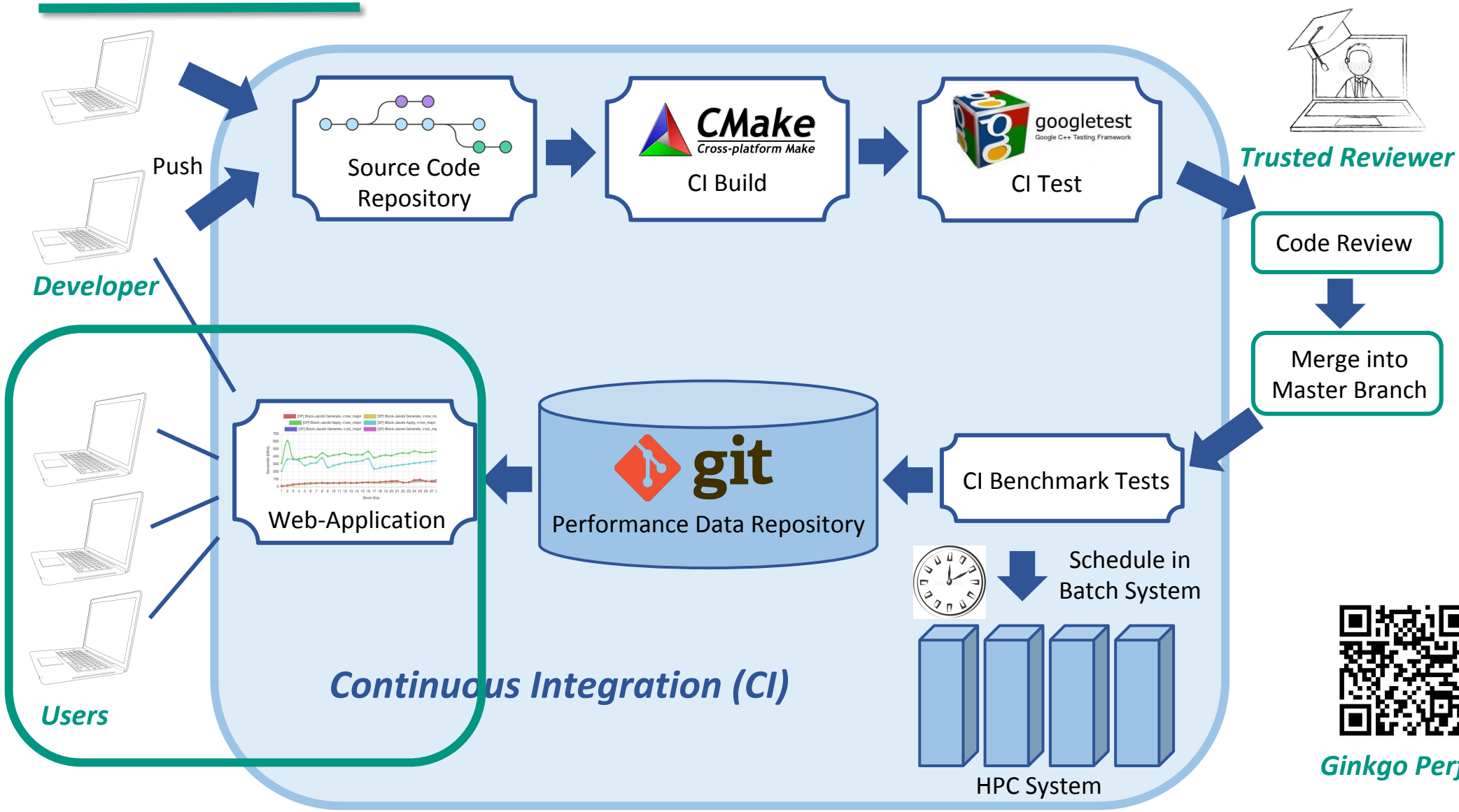
- Store data in separate repository.
- Use JSON data exchange format.
- Cloning a huge repository is expensive.
- *"Interest in my experiments, only."*

A Healthy Software Development Cycle



- Store data in separate repository.
- Use JSON data exchange format.
- Cloning a huge repository is expensive.
- *"Interest in my experiments, only."*
- Interactive web-based Ginkgo Performance Explorer (GPE).

A Healthy Software Development Cycle



GPE

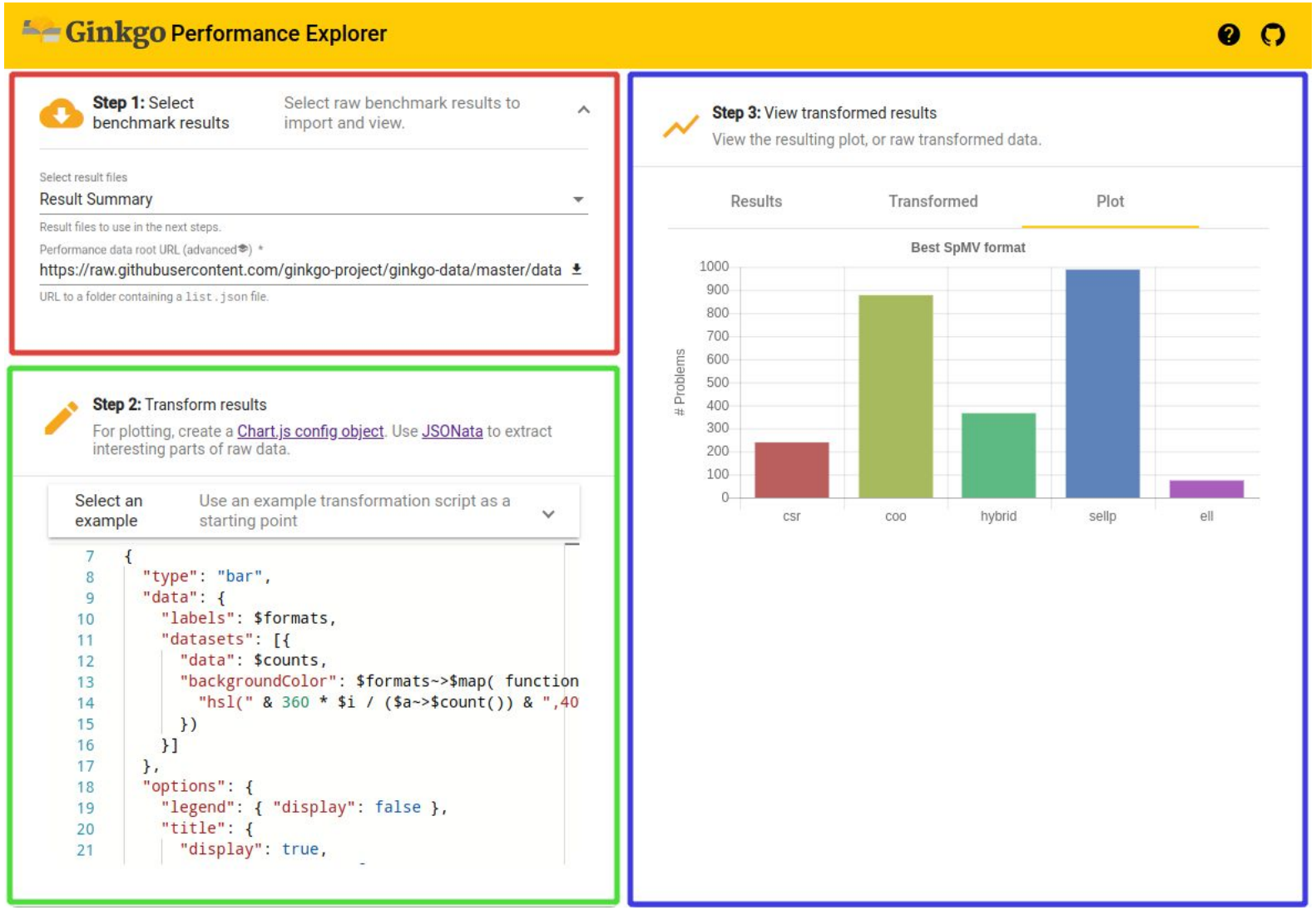
Ginkgo Performance Explorer

Ginkgo Performance Explorer (GPE)

Data Selection Tab

Transformation Script Editor

Data and Plot Viewer

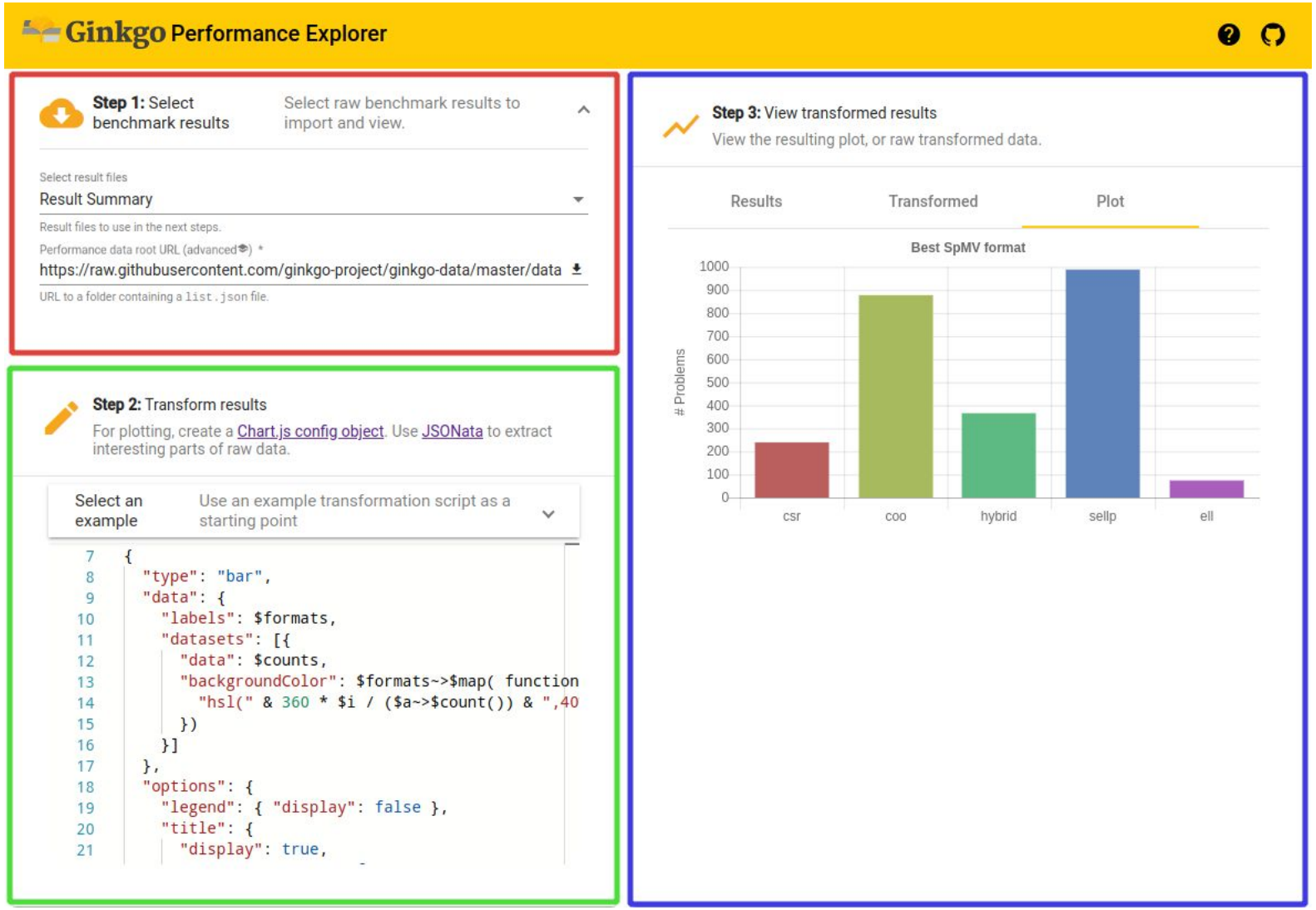


Ginkgo Performance Explorer (GPE)

Data Selection Tab

1.
Select Data in Git repository.

Transformation Script Editor



Data and Plot Viewer

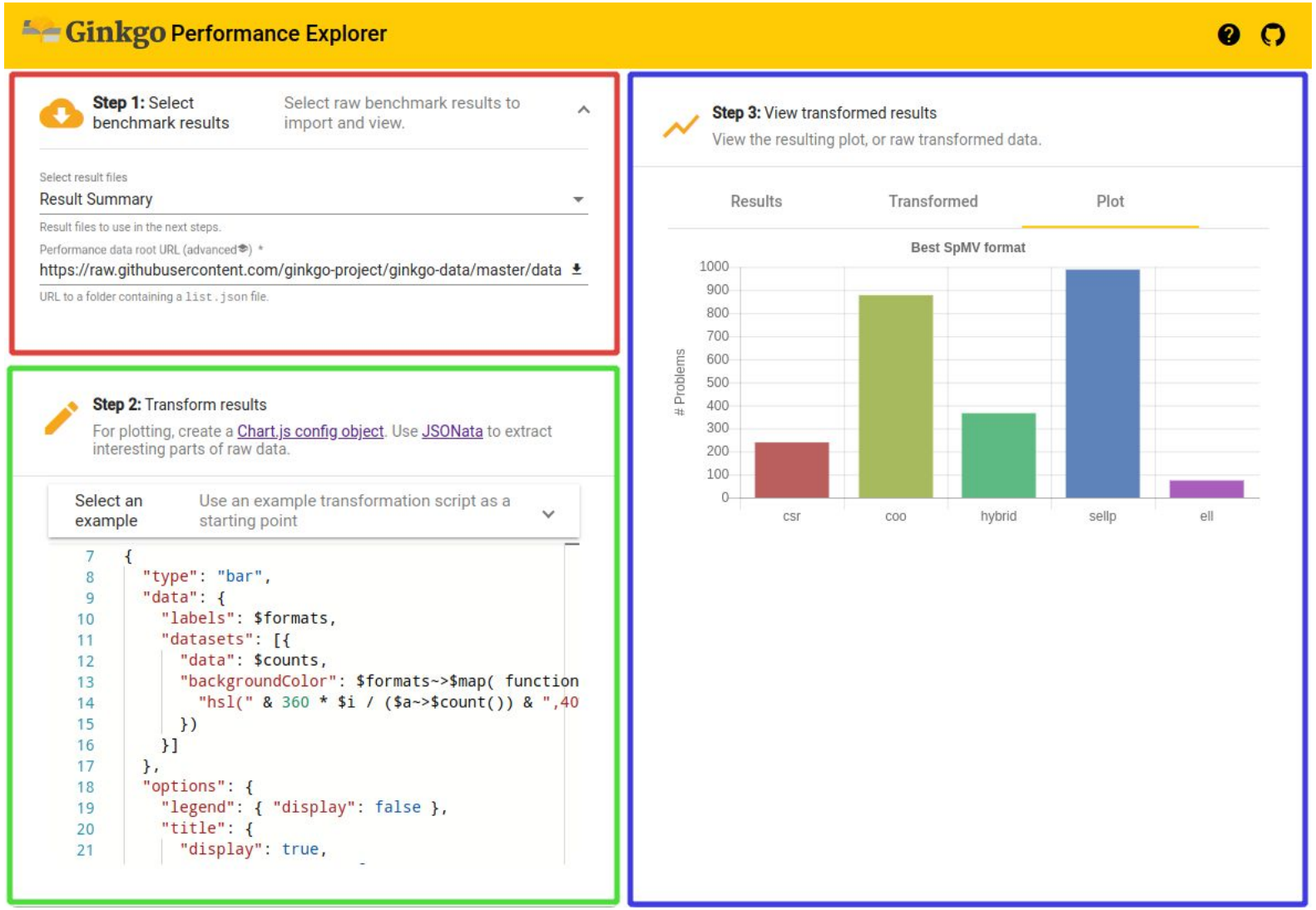
Ginkgo Performance Explorer (GPE)

Data Selection Tab

1.
Select Data in Git repository.

Transformation Script Editor

2.
Write JSONata script to visualize data (examples are provided).



Data and Plot Viewer

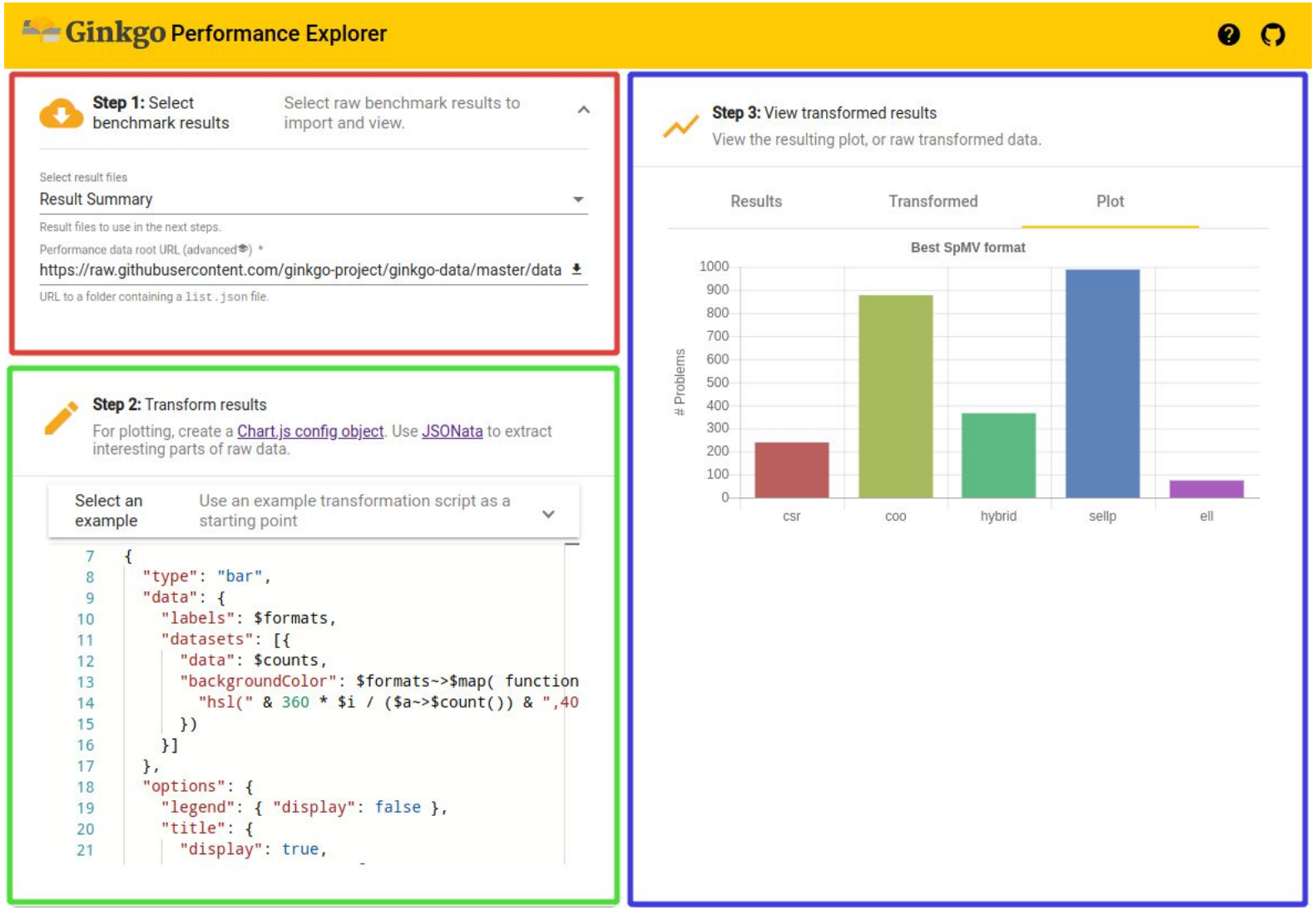
Ginkgo Performance Explorer (GPE)

Data Selection Tab

1.
Select Data in Git repository.

Transformation Script Editor

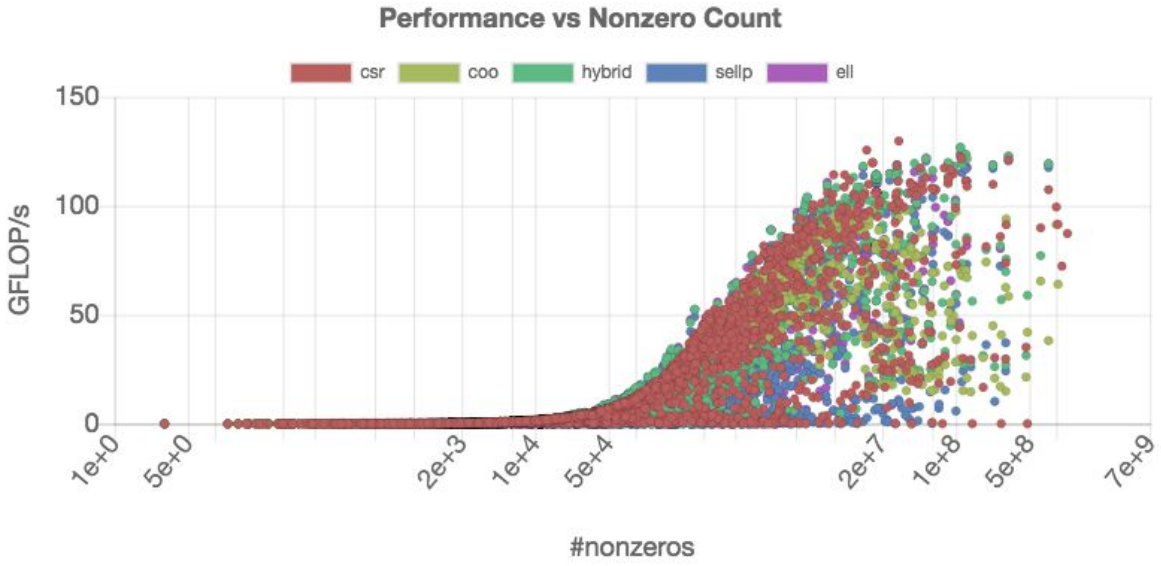
2.
Write JSONata script to visualize data (examples are provided).

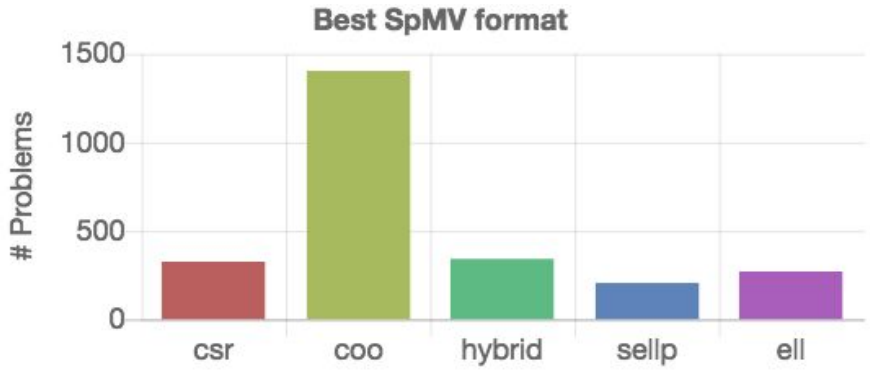


Data and Plot Viewer

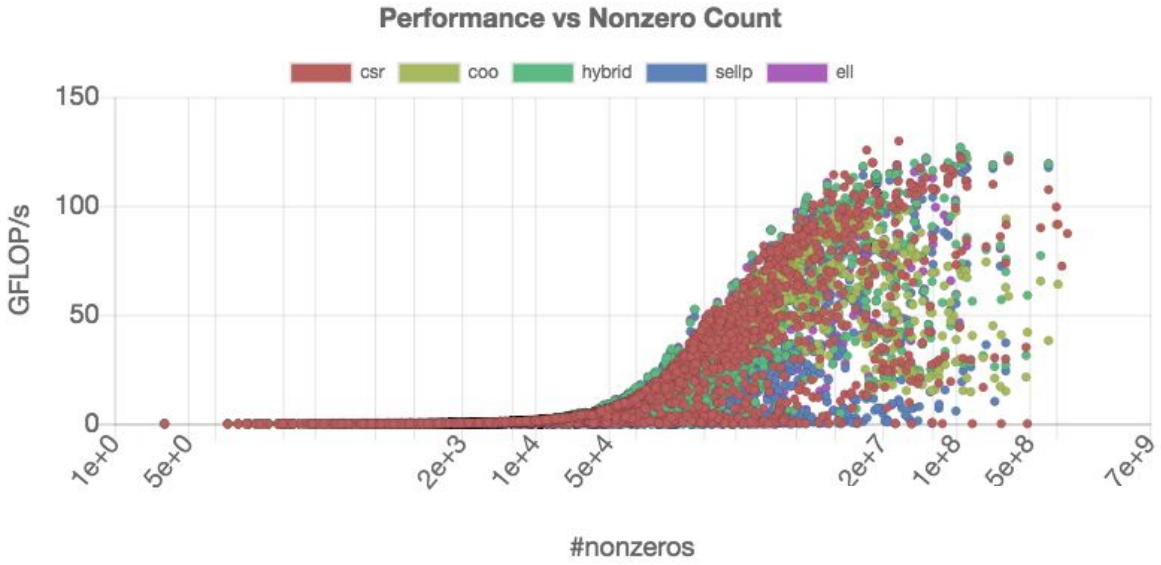
3.
Analyze data visually.

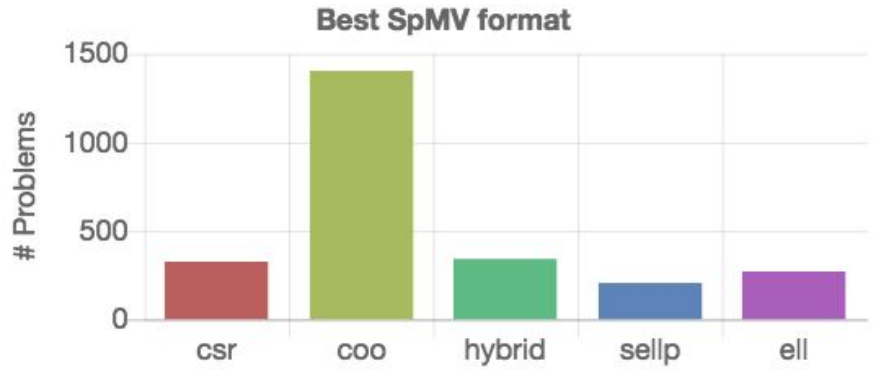
Ginkgo Performance Explorer (GPE)

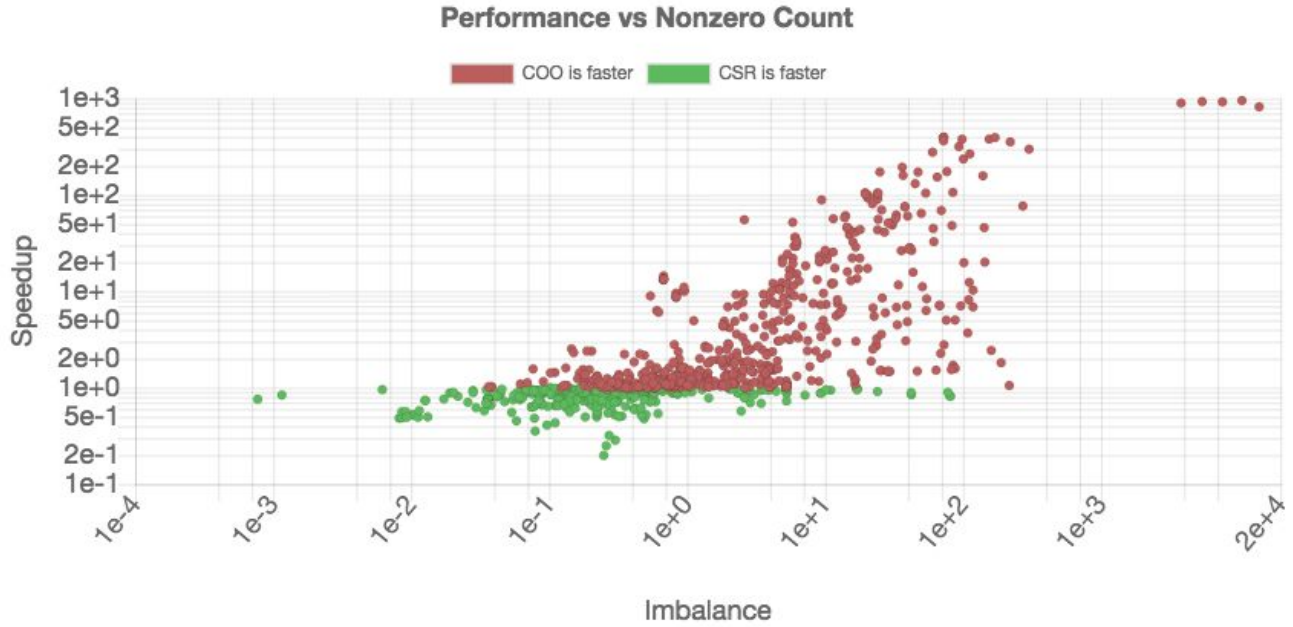




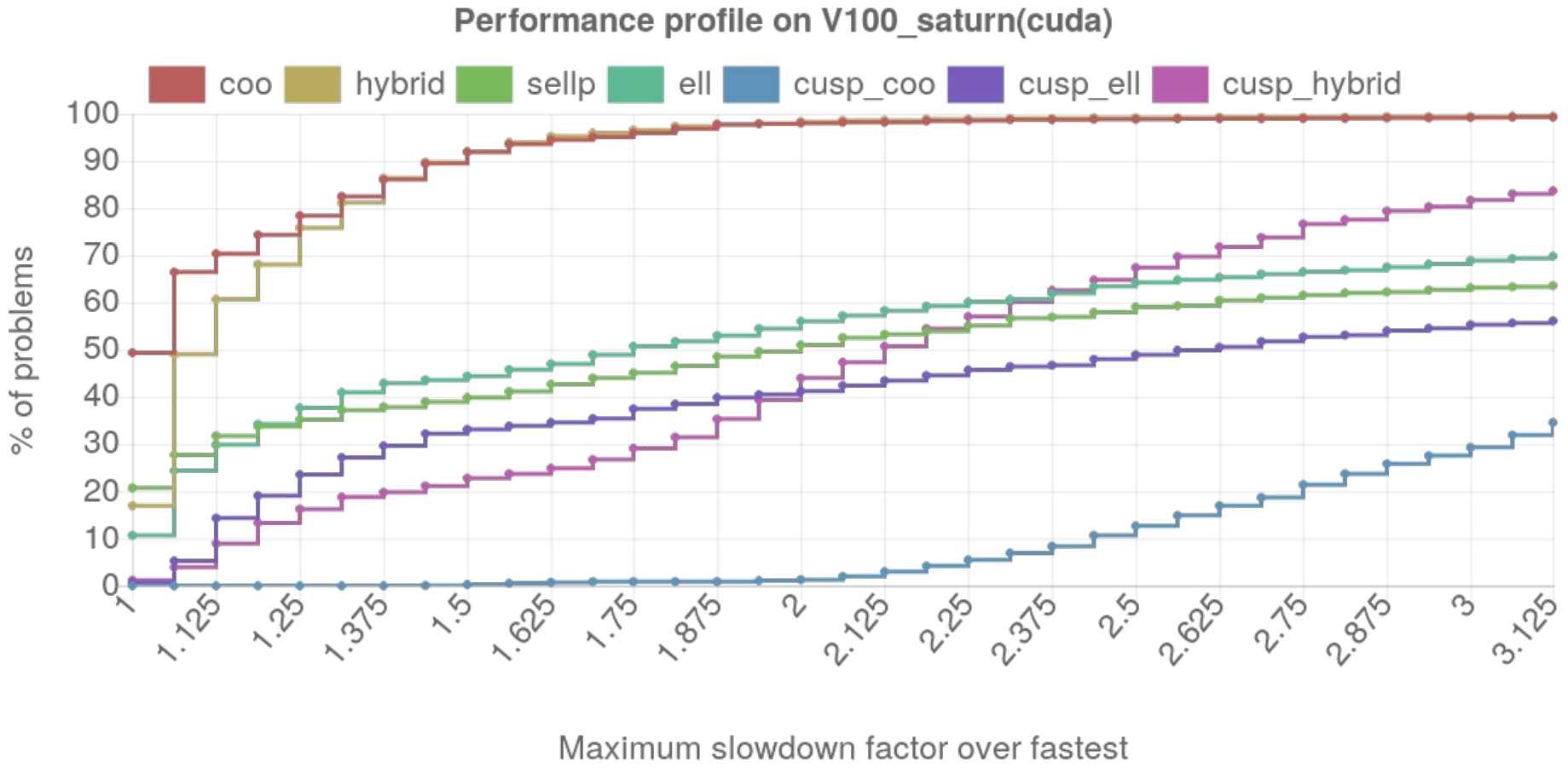
Ginkgo Performance Explorer (GPE)







Ginkgo Performance Explorer (GPE)



Dolan & More: Benchmarking optimization software with performance profiles

Ginkgo's Performance Evaluation Framework

Continuous Benchmarking Benefits

- Archiving performance data along with execution parameters ensures **full benchmark reproducibility**.
- **Comparing** the performance results over the code lifetime identifies **performance degradations**.
- **Ease of use**: the setup allows to launch benchmark with few clicks.

Ginkgo's Performance Evaluation Framework

Continuous Benchmarking Benefits

- Archiving performance data along with execution parameters ensures **full benchmark reproducibility**.
- **Comparing** the performance results over the code lifetime identifies **performance degradations**.
- **Ease of use**: the setup allows to launch benchmark with few clicks.

Ginkgo Performance Explorer (GPE) Benefits

- The design of GPE efficiently realizes the analysis as **web service**, removing the need for downloading performance data to local disk or installing additional software.
- **External developers** without access to HPC systems can test and engineer **their codes on HPC resources**.
- **Extensibility**: Option to **compare** performance with **other software** libraries.