Development of Auto-Validation System of BOSS

Di Jiang[1,2], Ye Yuan[1,2], Xiaobin Ji[1,2], Qiumei Ma[1], Yao Zhang[1]
[1]Institute of High Energy Physics
[2]University of Chinese Academy of Sciences

Motivation

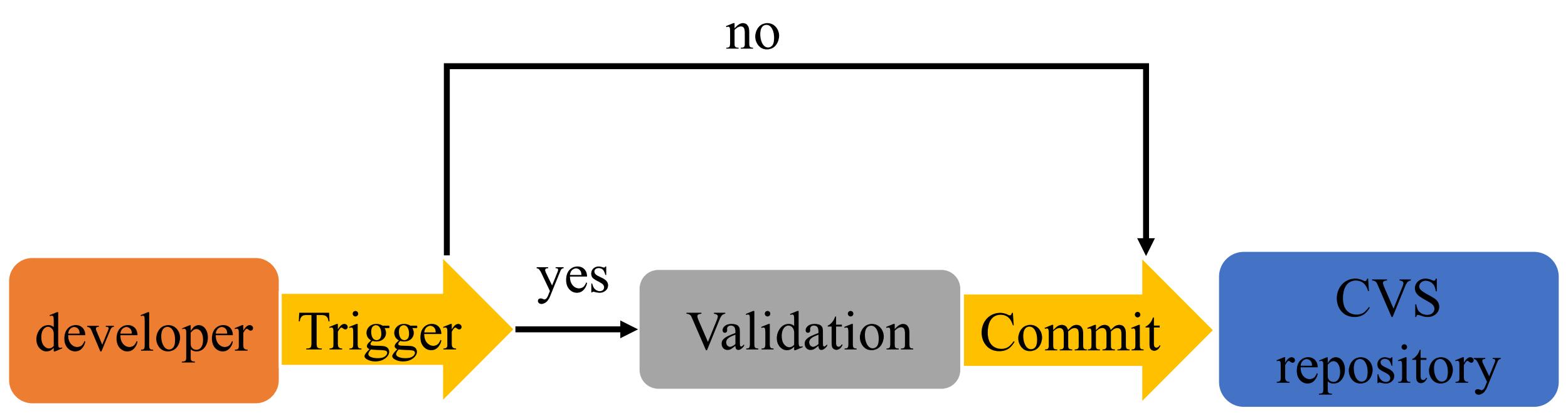
BOSS, which stands for BES Offline Software System, uses CVS as a version control system. As a result, BOSS is maintained and updated manually. However, modern development tools automatically maintain and update projects, including automatic compilation, validation, feedback provision, and real-time on-site release of new versions. In comparison, CVS appears to be very inefficient. Therefore, the version control system of BOSS needs to be upgraded. However, refactoring a large-scale operational system like BOSS can be challenging and difficult to implement. Changing established habits requires careful consideration. In the meantime, we can focus on upgrading our software development process based on CVS.

Auto-Validation System of BOSS

The idea is to provide an automatization script that is integrated with cvs to perform validation before committing changes. Here is an example command:

```
cvs commit –v 7.1.0.d –m 'test'
```

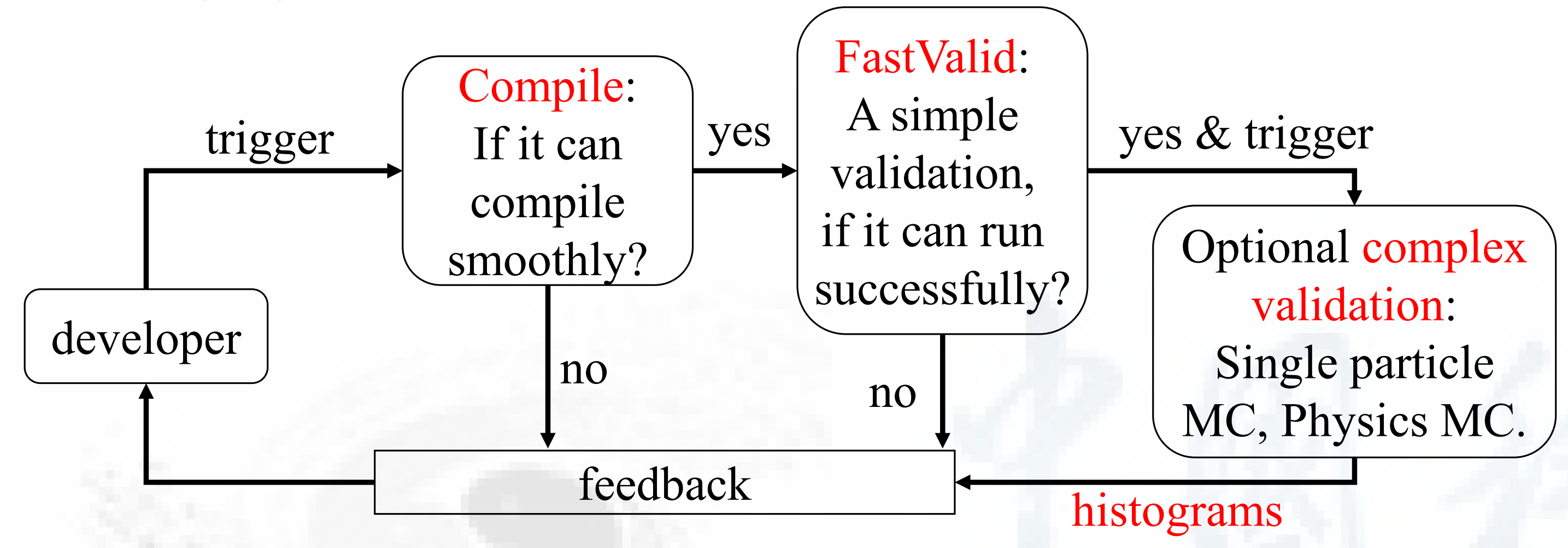
And the flowchart is shown as the following figure.



- Immediate feedback on validation results by accessing the standard output.
- Allows quick identification of any issues or errors.
- Give developers the freedom to choose whether they want to validate their changes or not.

Validation

The flowchart of the validation process integrated within that script is depicted in the following figure.


trigger
Compile: If it can compile smoothly?
yes
FastValid: A simple validation, if it can run successfully?
yes & trigger
developer
no
no
Optional complex validation: Single particle MC, Physics MC.
feedback
histograms

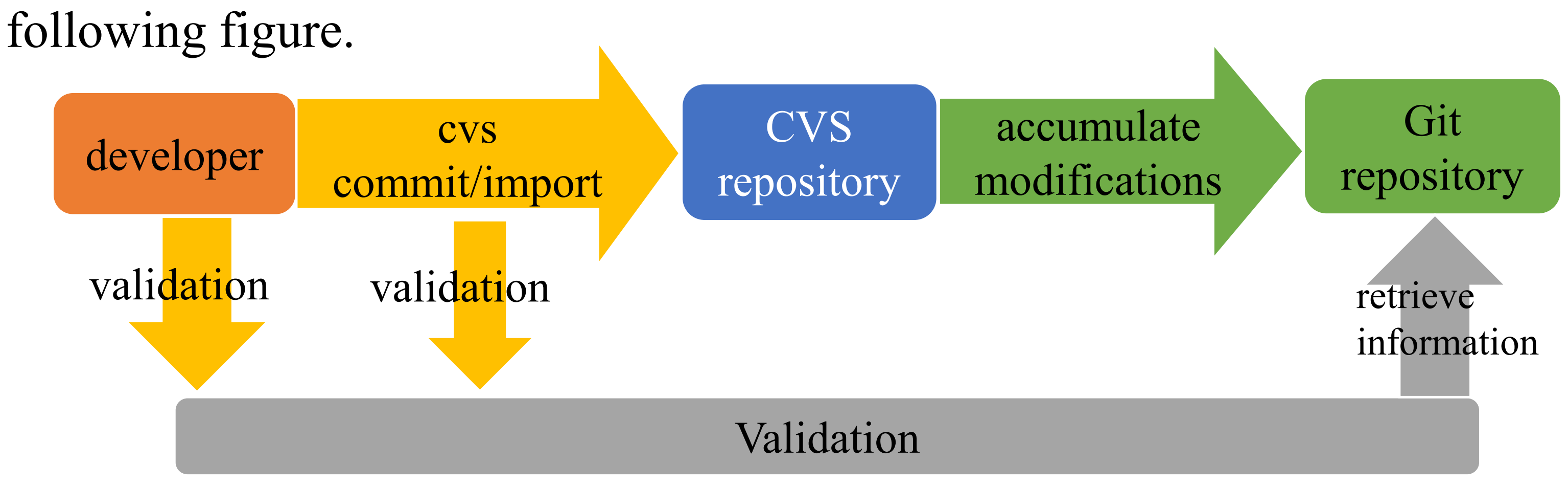
- If the code does not compile smoothly, the developer receives feedback.
- If FastValid validation fails, feedback is provided to the developer.
- Towards the end, two optional validations are carried out: single particle MC and physics MC. These validations do not reject any code but instead generate histograms that can be utilized for manual code evaluation.

Overall, the validation process integrates and automates the work of the Data Quality Validation Groups. And developers don't have to wait for feedback from the Data Quality Validation Group.

However, in cases where multiple packages need to be upgraded and validated, the current process would fail.

Multi-packages validation

When dealing with the validation of multiple packages, it becomes difficult for retrieving recent modification information from CVS. In such situations, utilizing Git as a cloud repository could offer a more effective solution. The flowchart is depicted in the following figure.



- The system can automatically transfer the content of each commit to the Git repository:

  https://code.ihep.ac.cn/jiangdi/BOSS
- Allows the recent modifications to accumulate in the Git repository.

To initiate the validation process for multiple packages:

- conduct the validation after committing all changes;
- perform the validation during the last commit itself.

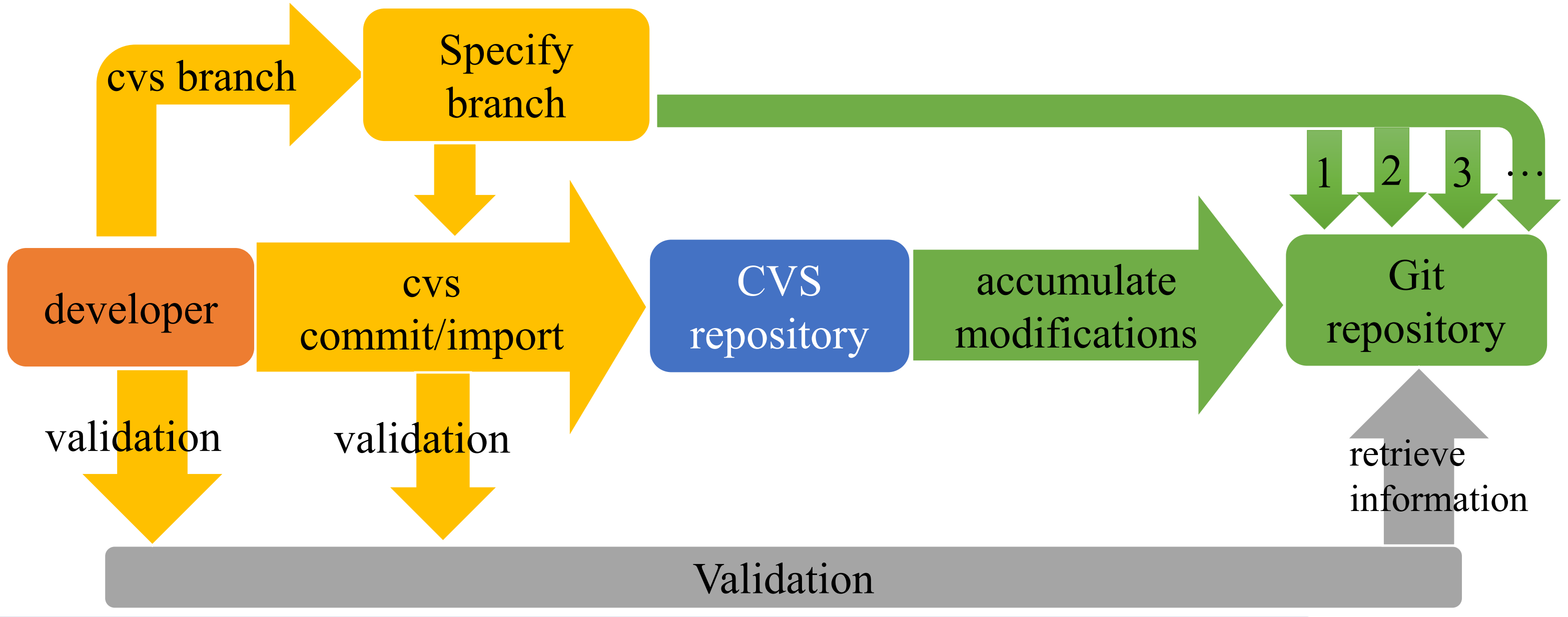
The validation process can then automatically access the required information from the Git repository.

Multitask validation

In practice, developers often handle multiple tasks simultaneously. To make it convenien, a 'branch' function has been developed.

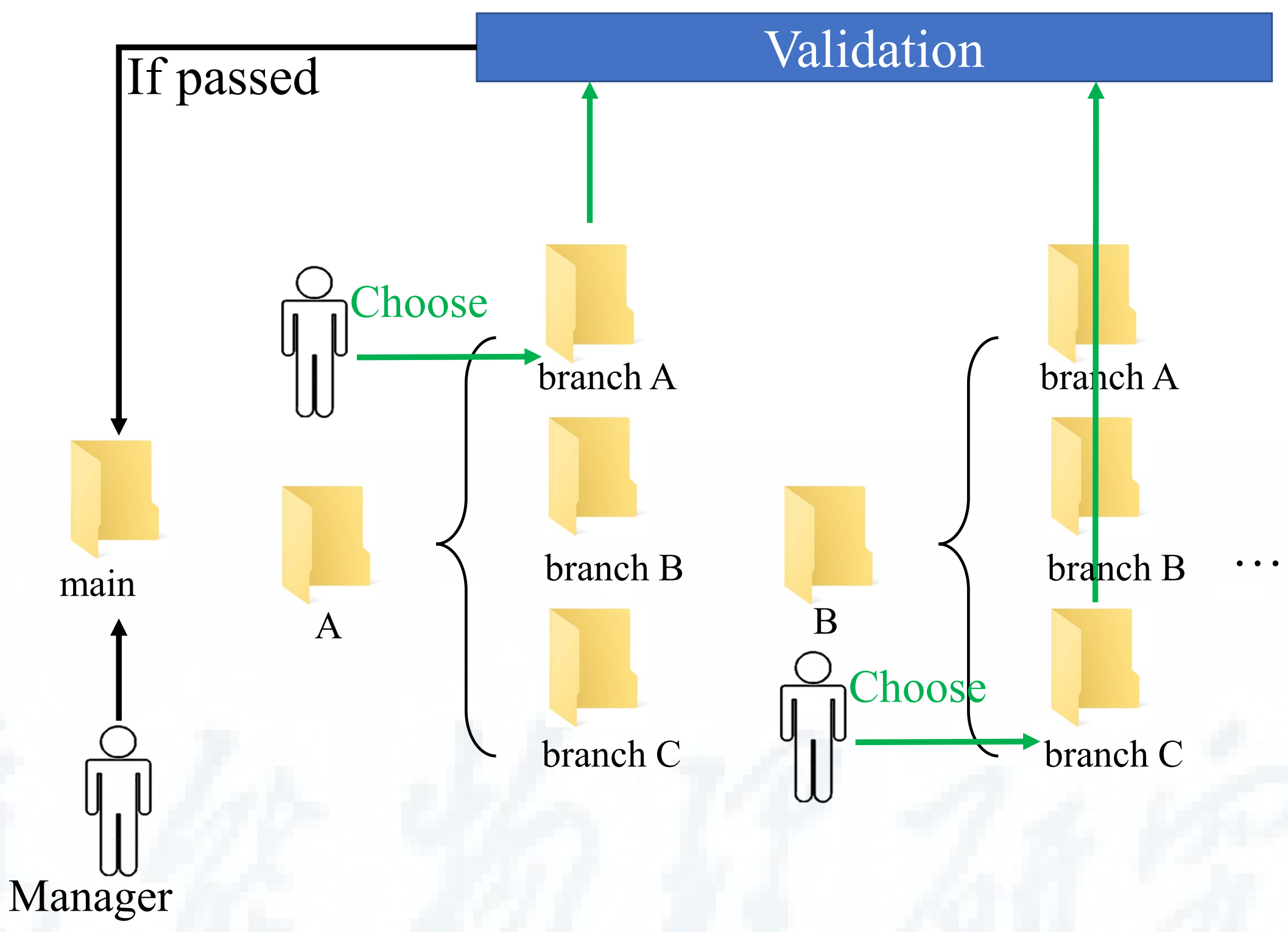
- Developers can store modified packages in personalized branch for a specific task.
- Packages within the same branch can undergo validation collectively.
- Developers have the flexibility to select any branch for validation as needed.

The final flowchart is shown in following figure.



Assistance of release of new version

To streamline the process for the BOSS manager, a main branch is created and managed by the manager for collecting validated packages automatically.



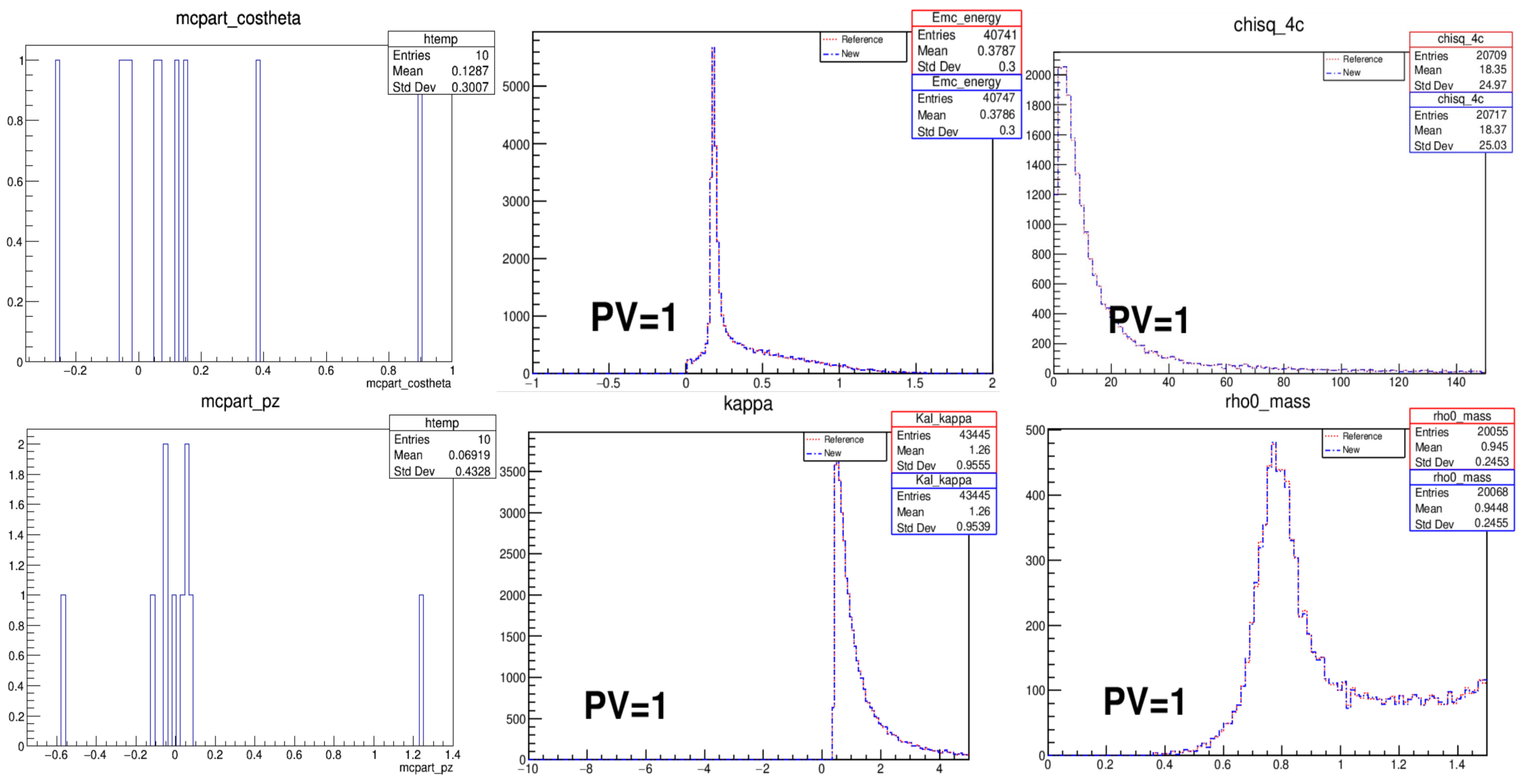
As shown in the above figure,

- Developers have the flexibility to choose their tasks for validation on separate branches.
- Once a task passes validation, its contents are automatically transferred to the main branch managed by the manager.

This streamlined approach eliminates the need for developers to individually inform the manager of their contributions and need of manually checking out packages.

Validation Result

- The system has passed α test. Participants are Yao Zhang, Qiumei Ma, Chunxiu Liu.
- Example results of FastValid(left two) and complex validation(right four).



Summary

- Automatic validation system based on CVS is developed.
- BOSS developers can enjoy the benefits of modern software development ideas.

User manual: https://ihepbox.ihep.ac.cn/ihepbox/index.php/s/yY4GsRO8kkMBQEI
Code: https://code.ihep.ac.cn/jiangdi/automatic-compilation-and-validation-of-boss