

**PHUSE EU Connect
2024 – DH12**

Validation and Open-Sourcing: Can We Tame the Two Headed Beast?

Ben Barnaby-Pass, PHASTAR



Overview

What's Going To Happen?



Breakdown

A breakdown of what this presentation is going to go over and a short overview of what each component consists of.



Validation

Brief overview of validation within open-sourcing and how it links in with the pharmaceutical world.



Open-Sourcing

How the pharmaceutical world is trying to open-source and how we can keep the landscape safe.



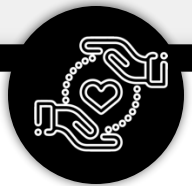
Pros And Cons

A look at what state the open-source landscape is in. Are we making more questions than answers?



How We May Tackle These Issues

How I think we should be able to iron these questions out.



Q+A

If you have any questions don't be shy by putting your hand up during the presentation!

Validation

What is the purpose to validation? And why do we follow this process? For instance, the FDA has no specific requirements for clinical studies; however, they do want to make sure that the data and data analysis submitted to them support the claims made by the sponsor.

SPECIFICATION

Before working on a table. We need to make sure that we know what the TFL is trying to outline. Once this is done, we can then start working on these TFLs and get a step closer to verification.

VERIFICATION

We then use this specification to create a TFL. We need to make sure we adhere to the information within the header, setup module, program comments and the naming conventions.

CONTROL

Once TFLs are complete, we need to keep the coding standards and ensure that the table continues to follow the validation steps. Ensuring constant tracking of updates.

QUALITY

If we follow these 3 steps throughout the whole process, then we ensure that the client and the governing body are happy with the quality that we are working at. It's a win-win!

Open-Sourcing

This is the hot topic now! Are we able to create packages and open-source them to clients and anyone that may require these packages? How are we able to do this and ensure validation is maintained?

01

Coding Language

Which language will we use?

The two main thoughts seem to head to RStudio and Python.

However, for every pro there is a con!

02

Existing Packages

If using RStudio, there are existing packages which we can use. This requires an extensive validation process.

03

Creating Packages

For this we need to make a package. We need to validate these processes and ensure we are controlling the package.

04

Validation

We need to ensure that the packages are doing as intended. This means using a frozen version of the package, only updating when necessary.

05

Distribution

When the package is finished and officially released, we need to make sure that the package is up to standards. This requires validation steps to pass.

Validation

Less Variation

Using a package helps to ensure that programs have consistency throughout. Creates an environment where we can spot mistakes.

Saving on Resourcing

Open-sourced packages help save on time and resources as we have the main function of the process already set up.

Traceability

We can keep records of updates and create solid version control.

Quality Control

Open-sourced packages which have passed validation have been checked to make sure quality is upheld,

Pros and Cons

Open-sourcing has its pros and cons. Outlining them helps ensure that we know our purpose while doing the process.

Maintenance

Outside packages need to be validated with new updates. The validation process takes a while and will slow down timelines.

Outdated?

Constant checks needed to ensure that each package has had updates

New Processes

On-boarding of talent or relevant training would be needed to ensure high quality processes.

Slow Processes

Validation is a big step in open-sourcing inclusion of a new package needs validating and this takes time.

How Can We Tackle These Cons?

After outlining some of the cons. There are 4 ways to be able to take these into account and try to avoid them as much as possible.

Freezing Updates

01 Doing this will ensure that the used packages are not updated and cause issues with the given programs when the package is open-sourced. Making sure we only unfreeze them once validation has passed on the packages.

Training/Onboarding

03 Validation and maintenance require a team. Training processes are needed to be outlined and distributed to ensure the team is all on the same page when working on the package.

Maintenance

02 Constant upkeep of packages is needed. Official release of a package needs a team to be able to iron out any issues or updates that a package might need. This ensures high level consistency with packages.

Version Control

04 This ensures we have a backlog of changes to keep a constant notice on what has been updated and keep accountability for issues that might arise when we end up validating the package. And if any issues come up, we can backtrack to past updates.

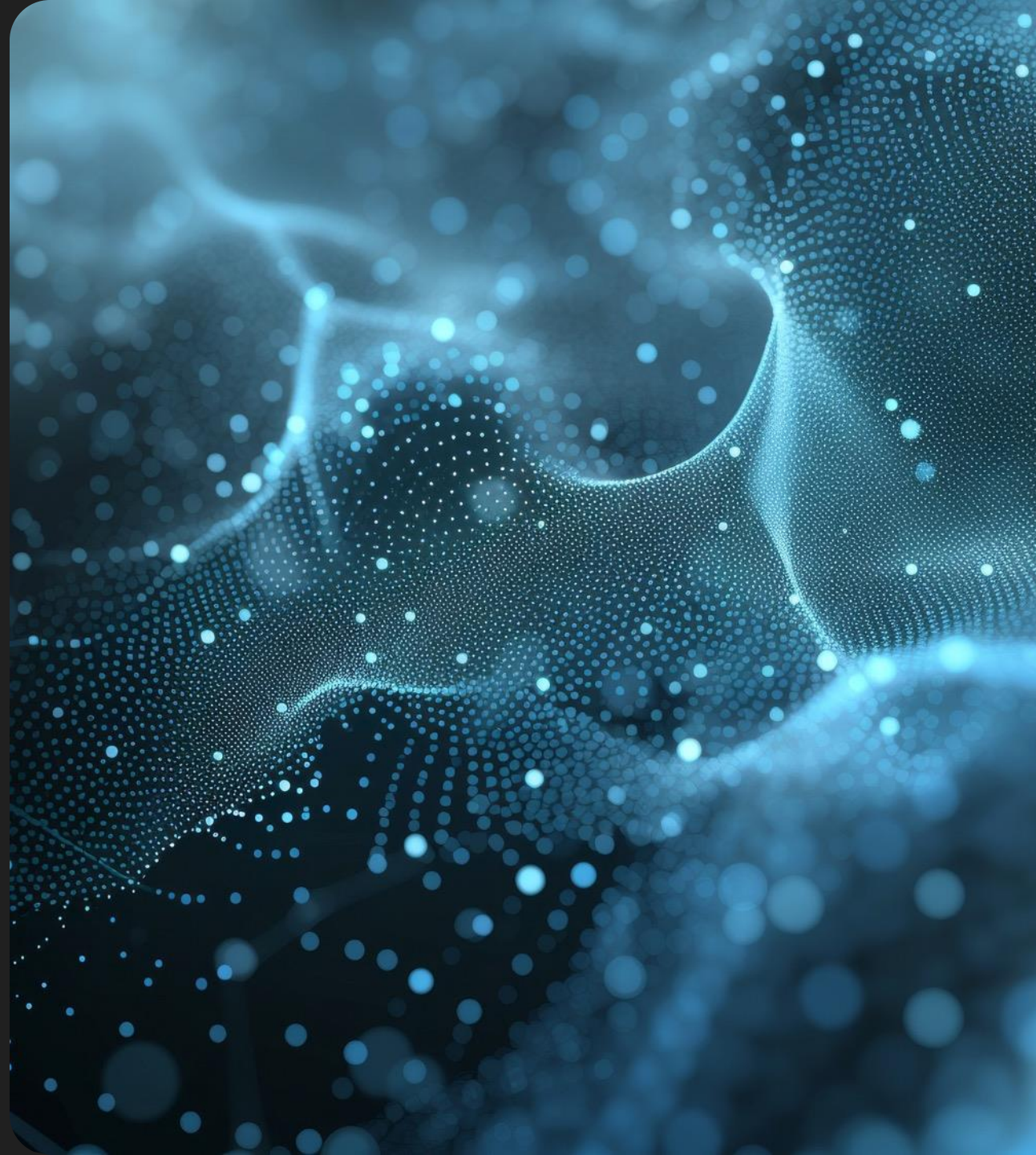
Thank You

Ben Barnaby-Pass

Phastar

WFH, UK

ben.pass@phastar.com



Q+A

Any Questions?