

# Streamlining the Validation Process

Discussions during the Knowledge Bazaar at PHUSE US Connect 2023

## Discussion highlights:

👉 **Validation tactics** - risk-based validation, specifications checks, spot checks, double programming, output crosschecks

👉 **Common issues** - unclear specifications, communication difficulties, versioning issues, and more

👉 **Possible solutions** - validation training, full independence between programmers, automated processes



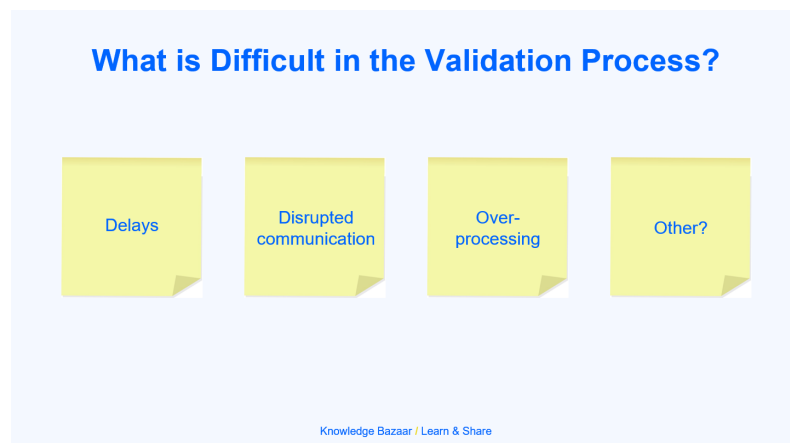
During the reporting process of clinical trials we create many outputs based on various input datasets and requirements. We use the input data, statistical analysis plan (SAP), mock shells, and Table of contents (TOC) to create and validate the output.

As the process of creating the output based on all this input is complex, many errors can occur during the process. Outputs can be based on a deprecated input, code errors could have been made, or requirements could have been interpreted other than expected.

Different stakeholders can do the validation within the company, like the clinician, medical writer, and a second programmer. At the knowledge bazaar, we focussed on the validation by programmers.

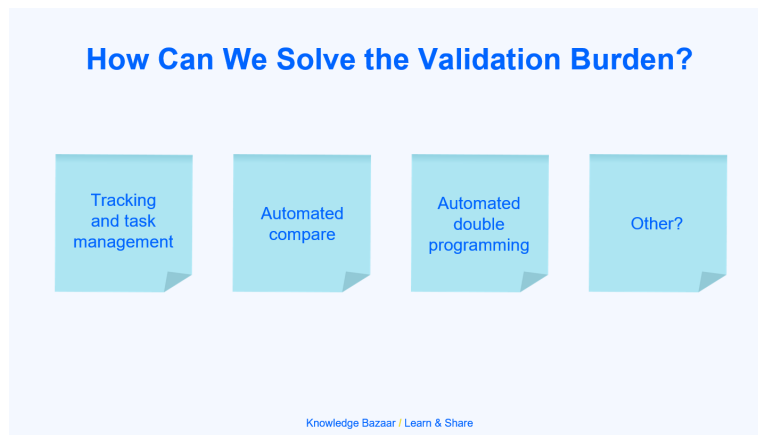
For them, we identified the following validation tactics:

- Risk-based validation: based on impact on patient safety the extent and detail of validation is determined.
- Control against specifications: like SAP, Mock shells, and Table of contents
- Spot checks: check the content of outputs and check outputs against source
- Double programming: re-program output by a second programmer and compare
- Output crosscheck: check the output of different tables, listings or figures against each other



With the group we discussed what is difficult in the validation process. They came up with the following issues:

- Unclarity in input specifications like Mock shells
- Variable name and type not clear or incorrect, resulting in output issues
- Sorting of data not univocal
- No metadata communication
- Issues in global communications
- Stability of data sources
- Program versioning issues



We also discussed how we could solve the validation burden. We identified that training in validation would help improve the output quality. Furthermore, we need to be aware that the programmers should be fully independent of each other. This is not always feasible as programmers work based on the same training, with the same instructions, and are interacting with each other as colleagues. Also, working with the same standard code is not entirely independent, which is a common practice.

Finally, we identified that the validation of figures was the most difficult to do. Nowadays, this is in most cases done manually by means of measuring the correct position of a data point, which is a burdensome process. To have an automated solution for figure validation would be of great added value.

**We thank all the participants for their contribution.**

Please contact us if you have any additional questions, or comments or like to learn more about our product.

Beaoncure's COO: [ilan@beaoncure.com](mailto:ilan@beaoncure.com)

Beaoncure's Customer Success: [noa.sharon@beaoncure.com](mailto:noa.sharon@beaoncure.com)