Validation Methods and Strategies for Presentation of Clinical Reports: The Programmers Road Map to Success

Vijayata Sanghvi, inVentiv Health Clinical, Princeton, NJ

Abstract:

Validation is one of the core tasks of programming with in pharmaceutical industry with in clinical reporting space. This is very important job roles which weigh more responsibility on programmer's knowledge and experience to make sure the end product meet the study design requirements. This paper will outline list of methods and strategies a programmer can adopt to succeed in qualitative submission of clinical reports before been released to respective stake holders with in an organization or external agency for review. Further this paper will outline on the detailed tasks and guidelines on what elements of output will be validated and how, through a QC plan check list.

Introduction:

Validation is a very precarious component to programming clinical trial analysis. In order for, effective validation is it essential that the programmer has an in-depth understanding of the data with which he or she will be working. If the programmer does not understand how the data is organized, or how the values for each variable is reasonable within each dataset or understand how the data should behave, the programmer will not be able to ensure that the final result of his or her programming effort is comprehensive or even appropriate. Hence, to be an effective programmer in the pharmaceutical industry, the programmer needs to understand the validation requirements and learn how to make the code do the bulk of the work so that the programs are competent as well as precise.
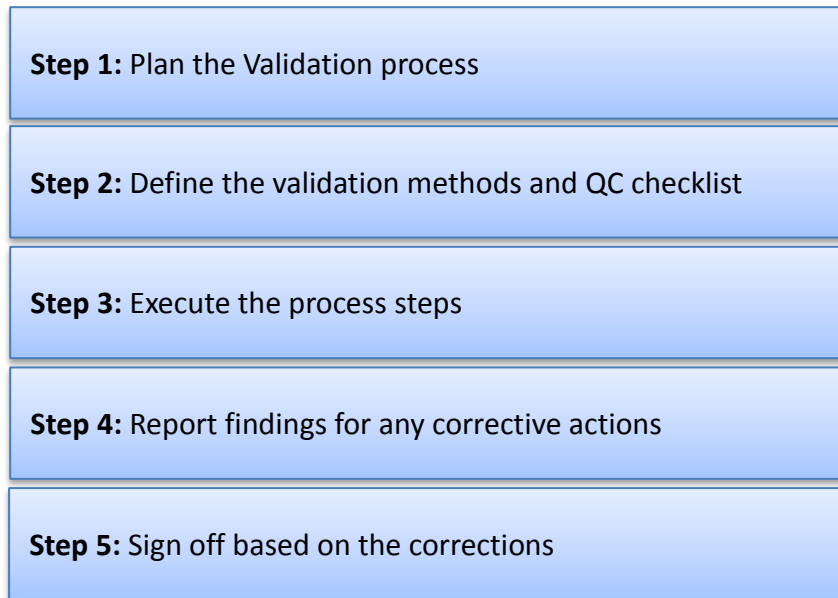
Validation process

Planning a validation process is a core task of validation work, regardless software been used. However, when you know that you might use a combination of software packages such as SAS and StatExact, the planning activity becomes even more important to consider the dependencies to work with. Key steps in planning include reviewing the SAP and mock TLFs and assessing software applicability, creating a validation plan that outlines activities.

It is essential to track validation steps in a systematic and logical manner since they are interdependent. Documentation of each step for instance, in the validation process is also crucial in capturing and verifying that the validation effort was done appropriately. Along with detail documentation, to capture the traceability of each validation task is important for the reviewer.

This paper will present methodologies that will make the process more organized and tractable. The goal is to add little or no effort from the user's perspective, yet gain the benefit of having a secured, audit trail of all steps performed during the development and verification process. This paper covers few of the aspects to give an overview of the process and methods which can be adopted based on the study needs and timeline in consideration.

Validation process flow

Step 1: Plan the Validation process

Step 2: Define the validation methods and QC checklist

Step 3: Execute the process steps

Step 4: Report findings for any corrective actions

Step 5: Sign off based on the corrections

Validation Methods

Double programming:

This is the widely adopted method of validation in the industry. The programmer creates independent code from specifications and compares its output with production.

Peer Review:

The reviewer will review the production code against respective specifications to confirm that the logic in the code is consistent with right source variables being used.

Output Review:

The reviewer will review output specifications to determine if program output has been derived according to specifications. The reviewer usually reviews only certain components of the outputs as part of the validation process.

QC Checks Frame Work across output types

General

- All tables, listings, and graphs defined in the SAP/ List of Tables are produced
- Tables, listings, graphs are compliant with SAP and Mock Shells
- All table numbers, titles and footnotes agree with the mock shells
- Treatment columns / labels agree with the request and ordered as per the mock shells
- All tables, listings, graphs contain reporting dataset creation and output generation dates
- Null reports are generated with a message, if there is not reporting data as per the criteria
- Page numbering and page breaking in the outputs are correct
- There are no errors or unexplained warnings or notes in any SAS log files
- All presentations are in acceptable format for user readability
- Variable names to meet appropriate standards
- Verifying all derived variables and handling of missing data

Design Checks

- Have the reporting datasets created as per the study design setup, especially it can get tricky for non-traditional study design

Treatment Assignment

- Have the treatment labels and column sequence been programmed correctly as specified in the SAP and Mock Shells

Source dataset checks

- All variables on the case report forms or external database transfer document are represented in the source dataset

Derived dataset checks

- All variables and datasets have explanatory names and labels as defined in the specification file
- Number of unique subjects should match to source data
- Check baseline values are derived correctly for findings domains
- Check to make sure any further deviations at variable level are done as per the study specification

Summary Reports

- Percentages are consistent with N of a respective treatment group
- Source data is footnoted
- The N counts are consistent across tables
- Summary Ns are consistent with supporting listings
- Mean values are consistent with category values/ranges
- Standard deviation values are consistent with ranges
- The decimal precision is been followed as per the reporting standards

Listings

- Verify all the source data is reported
- Verify the required variables as defined on the listings mocks are been reported
- The order of the variables is followed as per the mock
- The key variables are been sorted correctly
- The titles and footnotes are consistent to defined in the shells
- Group of variables of key variables is programmed correctly
- Date variables are formatted correctly

Graphs

- Verify it agree as per the mock shell provided
- Cross check to corresponding summary tables or listings for integrity of data reported
- Annotation of numbers on the graphs correspondence to right values as expected
- Verify the axis are reported to suitable scale as suggested
- Verify to make sure there is no underreporting of the data point to be reported
- Verify the program log files to look out for any warning and/or notes

Domain level checks

- AE
  - Have all AEs been coded as per the agreed MEDRA Dictionary
  - Check to make sure there are no truncations of preferred terms

- Lab's
  - Check to make sure all the lab conversation factors are available for respective lab tests
  - Make sure all the tests collected for the study are reported
  - Original units are available for the non-missing results if applicable

Conclusion:

Adopting a thoughtful process and pre-defined methods at report development stage to validate various types of datasets and outputs will enable an individual to deliver a quality product as per the study requirements. The above QC plan can be expanded based on the study needs with collaboration with the team.

Acknowledgements:

I would like to thank my employer inVentiv Health Clinical for giving me an opportunity to present this paper.

References:

Matthews, Carol I., and Brian C. Shilling, 2008. Validating Clinical Trial Data Reporting with SAS®. Cary, NC: SAS Institute Inc.

Author Contact:

Vijayata Sanghvi
inVentiv Health Clinical
Princeton, NJ
703-577-9136
Email:vijayata78@yahoo.com