

Software Validation in Clinical Trial Reporting: Experiences from the Biostatistics & Data Sciences Department

Andrea Baker, GlaxoSmithKline, Harlow, UK

ABSTRACT

Data reported from clinical trials must meet the highest standards of quality and integrity and comply with strict regulations from the FDA and other worldwide regulatory agencies. The FDA's acceptance of data from clinical trials is based upon its ability to verify the quality and integrity of the data during its onsite inspections and audits. This includes onsite inspection of documentation that demonstrates the validation of software which has been developed in-house for clinical data reporting.

At GlaxoSmithKline (GSK) we use SAS® software as the primary tool for the analysis and reporting of clinical trial data. This presentation highlights the need for software validation and describes how validation processes can be applied to clinical trial reporting.

The following topics will be included:

- Why Validate?
- What is Validation?
- What Validation isn't
- What Does Validation Involve?
- Implementing a Validation Process
- Conclusions/Considerations

COMPANY BACKGROUND

GlaxoSmithKline is a world leading research based pharmaceutical company. Its headquarters are based in the UK, with operations in the US. GSK is involved in therapeutic areas concerning respiratory, neurology, oncology, antimicrobials, metabolic, muscular skeletal, cardiovascular and urology. GSK Research and Development has over 16,000 employees based at 24 sites in 7 countries. The Biostatistics & Data Sciences (BDS) department is based in Europe and North America.

INTRODUCTION

This paper will provide an overview of validation and how validation processes have been implemented within the Biostatistics & Data Sciences department of GSK for its clinical data reporting software.

In the context of this paper, software is referred to as the analysis and clinical trial reporting programs generated within the Biostatistics & Data Sciences department using SAS® software.

WHY VALIDATE?

Overview

Validation is an integral part of the software development process and it is important to recognise that it does apply to the development of software used in reporting and analysing clinical trial data.

Quality and Reliability

Validation helps to demonstrate that a quality set of programs have been produced and will be maintained in a controlled environment. It helps to ensure that programming standards are being followed (e.g. through code review) and allows for the efficient re-use of programs.

Regulatory Requirements

Validation is also a regulatory requirement. Agencies such as the FDA will expect to see documented evidence of validation during their inspections and audits. These audits are conducted to ensure that data from clinical trials "meet the highest standards of quality and integrity and conform to the FDA's regulations."¹

ICH/GCP also specify that "All clinical trial information should be recorded, handled and stored in a way that allows accurate

reporting, interpretation, and verification"² and that "Quality control should be applied to each stage of data handling to ensure that all data are reliable and have been processed correctly."³

WHAT IS VALIDATION?

There are various definitions within industry documents on validation. Essentially the purpose of software validation is to demonstrate that it is performing its intended function accurately and reliably and that it will continue to do so consistently. To achieve this, validation needs to follow the processes of a system development life cycle and each stage should be documented, reviewed, approved and archived. For software already in use, validation will be retrospective and involve establishing that the software does what it is supposed to do through review of historical information.

WHAT VALIDATION ISN'T (Validation vs Quality Control of Output)

Quality control (QC) forms a critical phase for the Biostatistics & Data Sciences department, in ensuring that the results produced from the clinical trial data are complete, consistent and accurately reflect the data collected.

Various QC checks are performed and documented to ensure that:

- The SAS® program logs are free of errors, warnings etc.
- All derivations and algorithms are calculated correctly
- Data hasn't been truncated
- Adverse Event, Concomitant medication investigator terms have corresponding WHO/ATC codes
- Correct decimal precision is used
- Missing values have been handled correctly
- Data between related tables and listings match
- Outliers, illogical content etc. are investigated
- Report format and layout are correct

It is common for QC to be misconceived as being the same as validation and the words are often used interchangeably.

QC however, only provides confidence that for a particular set of data the program will consistently produce the expected results and that the raw data has been accurately reported.

Validation ensures that the programs used to produce the report or manipulate the data will perform the same function on all instances of data. It also involves the whole software development life cycle including design, implementation and maintenance of the program.

QC is all about data whilst validation is all about software.

QC does form an important component of validation, especially in the case of one-shot programs, but it shouldn't by itself be considered as validation or take the place of validation.

WHAT DOES VALIDATION INVOLVE?

Validation should follow the phases of a Software Development Life Cycle (SDLC):

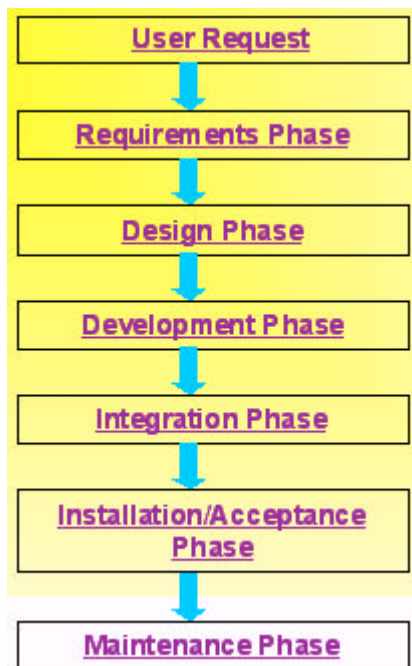


Diagram showing the flow of validation activities related to a SDLC.

The initial phase of a software development life cycle involves putting together a user requirement specification and a functional specification. The associated validation activities include writing a validation protocol, conducting a supplier audit (if the reporting software is custom built or bespoke) and completing a user requirement specification report. These activities are detailed below:

User Request

A validation protocol should be written to provide detailed descriptions of the procedures to be followed to validate the software and should include:

- Introduction and overview of the validation project
- Document history
- The organisation of the validation team
- Roles and responsibilities of the team
- Scope of the validation
- Development and validation strategy, including design , implementation, testing and maintenance of the software
- Acceptance criteria
- Certification and approval
- Periodic Review

The document should be put together by members of the validation team, reviewed by the validation project leader and reviewed and approved by a validation steering committee (the level of approval will depend on the size of the validation project e.g. a one-shot program for creating a report wouldn't require such a high level of approval as a system which generated randomisation codes)

Requirements Phase

A user requirement specification (URS) should be written which describes the requirements of the software in general terms. It should specify what needs to be done and not how it will be done. The URS should detail for example the input and output parameters of the program.

The URS should be written by the user and will need to be reviewed and approved.

A functional specification may also be written describing how the user requirements will be met and detailing those that won't be. It should also define the functionality that doesn't relate directly to the user interface (e.g. systems interfaces). Typically it is written by the developer of the software.

The URS and functional specification (if required) should be reviewed for technical content and quality. This review should be documented in a user requirement qualification report and should include details of the tasks undertaken during the review and version numbers or dates of the documents reviewed.

The next phase of the SDLC involves putting together a design specification. A design qualification report should be completed as part of the validation activities for this phase:

Design Phase

A design specification should be written which describes the software in sufficient detail to allow it to be programmed and should include detailed descriptions of algorithms to be used, any assumptions made and references to programming standards to be adhered to. For programs written to generate output for clinical reports, the design specifications may be included in a programming specification document. For one-shot programs, detailed comments in the code may suffice, but for programs shared across programming teams a separate design document would be necessary.

The design specification is reviewed in the design qualification report which may also include review of the functional specifications, flow diagrams and any deviations from the URS for the software developed.

Once the design phase is completed the next steps of the SDLC involve the construction of the code, software module testing and software integration testing. The validation activities associated with these phases include code and test script reviews:

Development Phase

During this phase the source code programs are written and code reviews of these should be performed. The code review should be undertaken by someone other than the developer for accuracy, clarity and adherence to programming standards. A line by line code review may not be practical, so in these instances a high level review using charts or diagrams may be performed instead. The review should be documented, detailing the date of the review, the name of the reviewer and details of any findings. The findings should be discussed with the developer and details of any actions taken documented.

Module testing will also form a part of the development phase to ensure that each individual module provides the correct output for a given set of inputs. These should be performed by the developer. Test script templates need to be developed for the module testing and these should be referred to in the Validation Protocol. The test scripts must be linked to the URS using a numbering convention to provide requirements traceability and should include details of the test actions to be followed, expected results, pass/fail indicator and comments relating to failed tests.

Integration Phase

This phase includes integration and system acceptance testing to ensure that all individual modules work together as one unit. For clinical trial reporting programs integration testing should check that a program containing calls to other validated macros works correctly.

System acceptance testing is the testing of the software's interfaces with other systems in the computing environment. Systems acceptance testing is probably not required for most clinical trial reporting programs.

Test scripts should be written for the integration and system acceptance testing in a similar format to those described for module testing.

The next stage of the SDLC is to perform the software installation and acceptance testing. Completion of an installation qualification, performance qualification and validation report form the validation activities for these phases of the SDLC:

Installation/Acceptance Phase

There are several validation activities associated with this phase. The installation qualification report demonstrates that the software has been installed according to the development groups specifications. Given the nature of most clinical trial reporting software, installation activities would be minimal and therefore the Installation Qualification may just consist of a review of the existing installation documentation, if it exists and several brief test scripts to test that the installation was carried out properly. Typical installation documentation may consist of actual installation steps and/or processes, software descriptions, operating parameters, and environment (operating system) requirements.

The performance qualification is a review of the software's performance in the production environment and should contain a presentation of the results of all the acceptance testing, a statement indicating if the expected results were attained, a description of any deviations from the expected results and a description of follow-up/revision activities and results. The performance qualification should refer back to the URS, the design specification, end-user documentation, system acceptance documentation and acceptance criteria for evaluation the tests results.

Training documentation and Standard Operation Procedures should be developed and reviewed at this stage along with procedures for change control and problem reporting.

A validation report should be prepared at this stage and include details of:

- What was done including a description of any deviations from the Validation Protocol.
- The results obtained from the Validation Protocol
- A summary paragraph indicating that the Validation Protocol was followed and the software met the validation requirements.
- The location of the documentation generated.
- A statement defining how long the Validation Report should be retained (the lifetime of the software and a follow-up archive period).
- A table of contents describing the individual documents that were compiled into the Validation Report.
- A Validation Certification sign-off page.

Validation certification is the last step in the validation process. The purpose of the certification is to review the Validation Report in order to verify that all validation deliverables have met the acceptance criteria that were described in the Validation Protocol.

The final phase of the SDLC is maintenance of the software and change control. The validation activities for this involve ongoing implementation of change control procedures and performing periodic reviews:

Maintenance Phase

Once the software has been certified as validated, change control and periodic review procedures need to be put in place.

Change control procedures are required to track problems and maintain a history of any changes made. They ensure that the software retains its validated status.

Periodic review is required to ensure that validated software is functioning correctly at pre-defined time points and should focus primarily on the change control log and problem log. The results of the Periodic Review should be documented in a Periodic Review Report. A determination of the validation status (still validated vs.

requires re-validation) should be the main goal of the report. If re-validation is required, then a Validation Protocol should be written detailing the validation activities required to bring the software back to a validated state.

IMPLEMENTING A VALIDATION PROCESS

Within GlaxoSmithKline, a set of company system validation guidelines existed, but it was recognised that these needed to be interpreted and streamlined specifically for use with the clinical trial reporting software used and generated within the Biostatistics & Data Sciences department. A validation team was formed whose goals included:

- Defining, standardising and documenting the program validation process for the department
- Drafting a set of validation guidelines for clinical reporting software which incorporated the company system validation guidelines and took into account regulatory requirements from agencies such as the FDA.
- Compiling an inventory of all software that needed validating
- Creating practical examples of how the guidelines could be implemented
- Consulting with regulatory compliance for advice and feedback
- Providing feedback on the team's progress to the department at various forums.

The Guidelines

The validation team was formed in March 2000 and a first draft of the guidelines was completed by August 2000. The table of contents shown below provides an outline of what the document finally contained. The appendices of the guidelines included templates for the various qualification reports.

Executive Summary

- ▣ 2. Identification and Assessment of Systems Requiring Validation
 - 2.1. Category Validation Details
 - 2.2. Assessment of Systems that Require Validation
 - 2.3. Determination of Validation Priority
- ▣ 3. Validation Team Charter
 - 3.1. Validation Team Name
 - 3.2. Membership
 - Team Sponsor :
 - Team Leader:
- ▣ Team Members:
 - 3.3. Role
 - 3.4. References
 - 3.5. Responsibilities
 - 3.6. Operations
- 4. Validation Protocols
- ▣ 5. Software Development Details
 - 5.1. SDLC Documentation
 - 5.2. User Requirements Specification
 - 5.3. Design Specification
 - 5.4. Source Code Review Documentation
 - 5.5. End-User Documentation
- ▣ All UNIX shells stored in phcommon should have a corresponding MAN(ual) page.
- 6. Design Qualification
- 7. User Requirements Qualification
- 8. Installation Qualification
- ▣ 9. Performance Qualification
- 10. Audit Report
- ▣ 11. Controls and Procedures
- ▣ 12. Validation Report
- ▣ 13. Periodic Review Procedure

The validation team split into sub-groups to draft text for each section of the guideline and several months were spent reviewing the text and amending as necessary. During this stage of development, various validation documents were referred to for guidance, including the company system validation guidelines, the FDA Guidance for Industry: Computerized systems Used in Clinical Trials¹, ACDM/PSI Computer Systems Validation in Clinical Research⁴, Validating SAS Programs in the Pharmaceutical and Medical Device industries by R.L Chamberlain and Joshua Sharlin⁵.

The draft guidelines were presented to a Biostatistics & Data Sciences Steering Team for review and comments. Mixed feedback was received:

- There was general agreement with the recommendations of the document and the importance of ensuring that the clinical trial reporting software was validated.
- Some of the validation activities (e.g. performance and installation qualification reports) were viewed as being too time consuming, especially for one-shot and adhoc reports.
- There was a sense of being overwhelmed by the amount of additional documentation required.
- The process seemed a lot of work which would require additional resources/time to achieve.
- There was some confusion over terminology e.g. what type of software did a 'system' encompass.
- There was difficulty in determining what documentation would be needed for various levels of software.
- Where would all the validation documentation be stored?

The Examples

The main concerns raised from the review of the guidelines were the volume of additional documentation required to perform the validation activities and difficulty in determining which activities applied to the various levels of software used within the department. Based on these concerns a decision was made to create some validation examples to help support the guidelines and demonstrate how they could be applied.

Two examples were chosen. The first was a program which was shared between reporting teams and performed weight, height and temperature conversions. The second example was a program which created a derived adverse event dataset from the raw data and was shared between studies within a reporting team.

For each example the various validation documents as specified in the draft guidelines were written and included the validation protocol, User Requirement Specification, Design specification, change control procedures, user acceptance test scripts, end user documentation, validation report and qualification reports for the user requirements, design specifications, performance and installation of the programs. Generally the documentation for the second example took less time to produce.

Once the examples were complete, they were presented to the Biostatistics & Data Sciences steering team for comments along with responses to the feedback received on the draft guidelines.

The general response to the examples was positive and approval was given to take the process further and proceed with two validation pilots.

The Pilots

The pilots involved lab flagging software and reporting environment programs (which were set up to standardise the directory structure and standardise and aid reporting processes across all the Biometrics teams) both of which were shared across reporting teams. Since the reporting environment programs were already in use, the validation for these would be retrospective. The version of the lab flagging software to be validated was newly developed so this validation would be prospective.

These two pilots were considered to be phase I pilots, but a phase II pilot was also initiated to look at validating within team standard clinical trial reporting macros.

The aim of the pilots has been to investigate how much time and resource is required to apply the validation process to various levels of reporting software and to find any 'holes' in the department validation guidelines.

These pilots will be completed by end of May 2001.

Outcome of the Pilots

Pilot1: Lab Flagging Software

Validation activities are still ongoing for this item of software.

Pilot2: Reporting Environment Programs

The SAS elements and shell programs that establish the department's reporting environment were validated and the department validation guidelines were followed very closely. It took approximately 200 person hours to complete the whole cycle of validation activities and this time included both learning and execution of the validation process.

Key Deliverables from the Pilots

The pilots have ensured that the department now has fully validated key software. The validation guidelines have been revised based on "real" experiences and there are now a number of department representatives with a higher degree of validation experience who can help advise and train other staff.

Learnings from the Pilots

There is a need to perform a risk assessment / cost benefit analysis when considering validation plans and this is especially true when validating one-off reporting programs. There is a significant learning curve involved when performing validation activities for the first time and this needs to be taken into consideration when the initial validation project plans are put together. Writing comprehensive user/functional and design requirements is a challenge and a substantial amount of time needs to be invested in making these as comprehensive as possible.

An important learning from the pilots has been the need for a dedicated validation team leader and an allocated time commitment from other members of the validation team. Delays to the pilots were invariably due to project work being prioritised over the validation efforts. Management needs to support validation and be willing to prioritise these activities. They also need to understand the benefits validation can offer. It is also important to have dedicated internal validation resources available to consult advise and train.

CONCLUSIONS

Validation is an integral part of all clinical trial reporting, regardless of how adhoc the report. Implementing a validation process is initially time consuming, but in the long term validation should ensure that we have programs in place which are reliable and can be re-used. There is a need to change attitudes towards validation as it is perceived to involve a considerable amount of time and resource and this has been one of the aims of the pilots. Validation is common sense, much of the process is probably being performed, but just not documented. An important point to remember is that validation is a regulatory requirement and documented evidence must be readily available for regulatory inspection and audits.

CONSIDERATIONS

Another goal after is to develop a set of SOPs and IOPs to enhance the validation process. e.g. two validation protocols could exist as IOPs, one to cover programs used within reporting teams and the other to cover more generic programs shared cross reporting teams. This would help to minimise the documentation required, whilst ensuring that a consistent development and validation approach is followed.

Although validation seems to involve a lot of work and putting a process in place can be difficult, it is worth considering that if a problem is caught early in the software development phase, it is far less work to correct than if it is found at a later stage e.g. just prior to submitting output to a regulatory agency. A validation process should ensure that critical problems are found and resolved at the earliest opportunity.

A pilot for one-shot programs needs to be initiated as there are time and resource issues to address with validation of these types of programs. It is probably even more crucial that these programs are validated due to the critical nature of these requests and the probability that the information will be submitted to a regulatory authority. It is possible with these programs that if a process is in place for quality control that this may satisfy most of the validation requirements.

There is a need to develop a central database to store the validation documentation

Another consideration is how are other pharmaceutical companies handling validation of clinical trial reporting software?

ACKNOWLEDGEMENTS

SAS is a registered trademark of the SAS Institute Inc. in the USA and other countries

REFERENCES

¹ The FDA Guidance for Industry: Computerized Systems Used in Clinical Trials, April 1999

² ICH/GCP Guidance for Industry, E6 Good Clinical Practice Consolidated Guidance, Section 2.10, April 1996

³ ICH/GCP Guidance for Industry, E6 Good Clinical Practice Consolidated Guidance, Section 5.1.3, April 1996

⁴ ACDM/PSI Computer Systems Validation in Clinical Research A Practical Guide

⁵ Validating SAS Programs in the Pharmaceutical and Medical Device Industries by R.L. Chamberlain and Joshua Sharlin

CONTACT INFORMATION

Your comments and questions are welcomed.

Please contact the author at:

Andrea Baker
GlaxoSmithKline,
New Frontiers Science Park (South),
Third Avenue, Harlow, Essex,
CM19 5AW. UK
Phone: 01279-646942
Fax: 01279-644430
Email: andrea_2_baker@sbphrd.com