

Validation of Programs developed Using SAS

Nikos Tsokanas, Roche Ltd, Welwyn Garden City, United Kingdom

ABSTRACT

A validation approach on programs developed using SAS, that satisfies the regulatory requirements, is beneficial to drug development businesses. The Food and Drug Administration (FDA) agency clearly states, within *21 CFR part 11*, that drug organizations must ensure the ability of their computerized systems to meet the applicable predicate rules. This is achieved by having in place a specific, verifying and controlled procedure, which needs to be documented, for all the SAS generated programs. In conjunction with a risk based approach, validation can be less time consuming, more effective and add more value to the overall process. High, Medium and Low Risk outcomes determine the relevant validation method that is then applied for the completion of the procedure.

This paper will demonstrate the keys to an effective validation process and expand with a high level overview of such an efficient approach used within the Biostatistics department in Roche. It will also highlight how the risk assessment is embedded within the process and the value it adds.

INTRODUCTION

FDA's main function is to ensure that the health of the public is safe. As far as the pharmaceutical industry is concerned, all the drug products must be safe, effective and of high quality. In order to guarantee health safety, the FDA had to generate a number of regulations and oblige all pharmaceutical businesses to be compliant with them. More specific, the regulatory authority is highly concerned with all the computerized systems that are used to generate information. In Biostatistics, the ambiguity in programming requirements and the possibility SAS programmers can make errors can sometimes generate misleading results in terms of analyses.

The FDA has issued a number of documents in order to "provide guidelines to persons who, in fulfillment of a requirement in a statute or another part of FDA's regulations to maintain records or submit designated information electronically " [1] Moreover, the creation of the CFR 21 Part 11 emphasizes the need of a validation approach to satisfy the regulatory authorities for any computerized system that is used to store record or electronic signatures [2]. However, these regulations are interpreted differently across the health organizations and cause the validation of clinical analysis to be time consuming.

The validation of SAS generated programs does not mean that a programmer needs to generate extensive documentation. It really means that the programmer needs to prove that the generated code is specific, verified and can be controlled. Within the Biostatistics department in Roche, the process for the validation of SAS generated programs is an essential concept for justifying both the FDA and Roche's business requirements. A risk based approach is used to validate (document) the activities SAS programmers complete.

THE VALUE OF VALIDATION

It is apparent that the FDA's concerns are reasonable and pharmaceutical industries are required to ensure that the data produced have to meet certain fundamental elements of quality for public safety. The validation of computerized systems used to generate data seems to result in an agreement between both the regulatory authorities and the health industries. The more accurate, informative and simpler such as a validation procedure is the better for both parties. Ensuring the quality of a piece of work is the quickest way to move forward a 'drug' project.

In biostatistics, despite full understanding of user requirements by all programmers, the hardest part is to write programs using SAS in order to generate hundreds of megabytes of datasets along with decades of outputs to meet each project's requirements. It is also difficult to guarantee the correctness of all this work if there is no evidence that testing has performed and all the work is validated. However, the process becomes easy by summarizing the most essential activities of the analysis and testing on a few pages as evidence that validation has been completed.

More specifically, when writing a program using SAS, the term 'validation' refers to the process needed to demonstrate that the SAS code under consideration meets the requirements in all respects. The approach adds value rather than creating extra work. For example, programmers using a documented validation approach can increase

PhUSE 2007

the usability and reliability of the SAS codes, which means that the same program can be used across a number of projects. This will result in less chance of error occurring and the overall time spent on programming reducing.

The level of detail in the validation should be limited to the point. The overall effort should commensurate with how much risk is involved in the developed program and should be seen as an interactive process. For example, the complexity of the program, what output the program needs to produce, etc, should be taken into consideration when defining the risk of a program. By all means, programming teams may change the risk during any stage of coding to define the process interactive, as mentioned. Hence, the risk is a key factor for the overall strategy of validation.

RISK MANAGEMENT

Risk is the combination of the likelihood of occurrence of a hazardous event and the consequence of that hazardous event. In the pharmaceutical industry the main area of risk lays on the *safety of the patients*. This risk is associated with development, manufacture and distribution of products which are safe and effective. Nonetheless *regulatory compliance* and *business process* risks should also be taken into account. Regulatory compliance risks involve any business process that is governed by local and national regulations (e.g., marketing, financial, manufacturing) and business process risks are those that are associated with the operations that enable achievement of the business goals.

Since risk is essential for all health organizations, pharmaceutical industries must manage this risk with the best possible method. A risk management strategy is highly recommended since it is a continuous process that involves the recognition of risk, the assessment of risk and the development of strategies to manage risk.

The term 'Risk Management' was first used by the FDA in August 2002 when the initiative, *Pharmaceutical cGMPs for the 21st Century: A Risk-Based Approach* was announced. The announcement describes that "the initiative seeks to integrate quality systems and risk management approaches into the existing programs and encourages adoption of modern and innovative manufacturing technology" [3]. Later on, the FDA published five initiatives in order to increase public safety for new medical products. Risk management was one of the five initiatives and encompasses a search for new and better ways to reduce risks. In addition, another document published by the FDA mentions the term 'Risk Management' indicating that pharmaceutical companies can choose to follow several types of validation practices, but the key was that the reasoning for the decision needs to be formally documented.

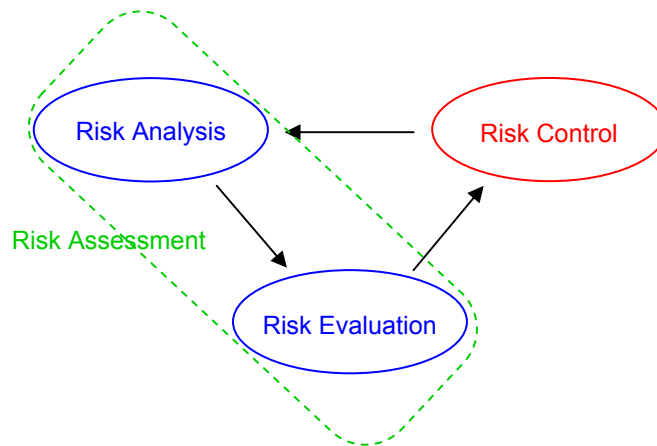


Figure 1 represents the main steps in Risk Management

There is a number of risk approaches that one can use, however, most of them are very complicated and time consuming. In SAS programming, a simple, effective risk method should be applied to ensure that public health will be safe with the information that is generated. When people manage a risk, they should really focus on three areas; risk analysis, risk evaluation and risk control. As Figure 1 illustrates, the risk assessment consists of two steps; risk analysis and risk evaluation.

AN EFFECTIVE VALIDATION APPROACH

An easy and effective approach to validate the statistical outputs (i.e. VADs, TFLs, etc) that are produced by using SAS has been employed within the Biostatistics department in Roche. The approach, Figure 2, focuses on generating an organized structure, which summarizes all the activities of each study (Reporting Event) in a single document. It

PhUSE 2007

helps people to easily navigate through the analysis of a study and respond to questions they may come up with at any stage of the process. As mentioned above, this validation method uses a risk-based approach to reduce the time spend to test that the reporting outputs successfully meet user requirements. Finally, it helps to identify low risk programs, which do not need as much validation as others within the same project.

The Roche SAS validation approach uses three different risk assessments on; the reporting event, the outputs and the SAS programs individually, in order to help programmers to understand their work and save time where possible. Furthermore, the validation approach is also split in five stages for simplicity purposes, which altogether cover the areas recommended by the FDA, *specification*, *verification* and *control*.

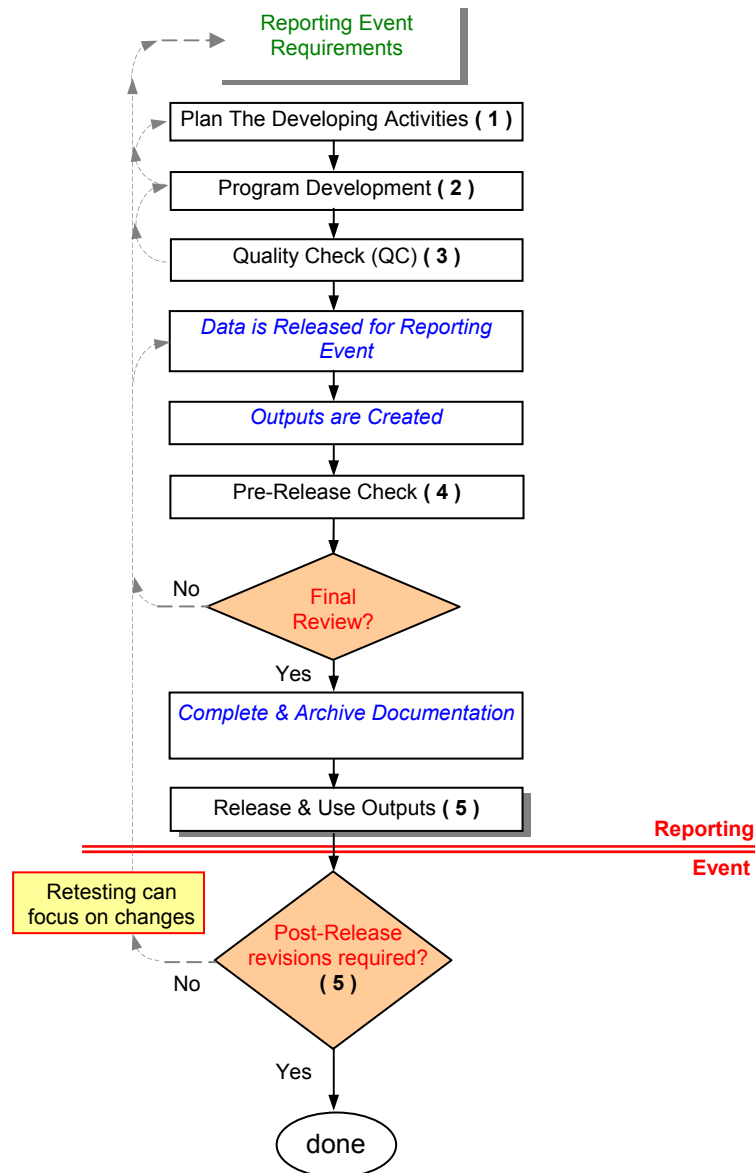


Figure 2 represents the validation approach used within the Biostatistics Department in Roche

1. Plan the Developing Activities

Initially it is important to make sure that certain actions have been taken to allow the programming team to start working. The user requirements which describe all the outputs of the reporting event must be approved by management. Each programmer has to be happy with and understand these requirements. The document to summarize the validation needs to be created and archived in a place accessible by all. It is important to update this document throughout the study with the main activities of the reporting event. The very first information captured is a

PhUSE 2007

brief description of the reporting event, the place where all the programming activities should occur and any conventions decided by the team, i.e. communication plan, how the testing and the change control will be recorded, etc.

Furthermore, the first risk of this validation process should be defined on the whole reporting event. This is an essential procedure as a considerable amount of time will need to be spent on verifying the outputs, which will be based on the definition of the reporting event risk. It is the decision of both the Lead Programmer and the Lead Statistician to define whether the risk of the Reporting Event will be high, medium or low. Such a decision need to take into consideration several factors such as; (i) the intended use of the outputs, (ii) audience (iii) the criticality of the data (iv) the formality of the analysis and (v) the programs that are used to produce the outputs.

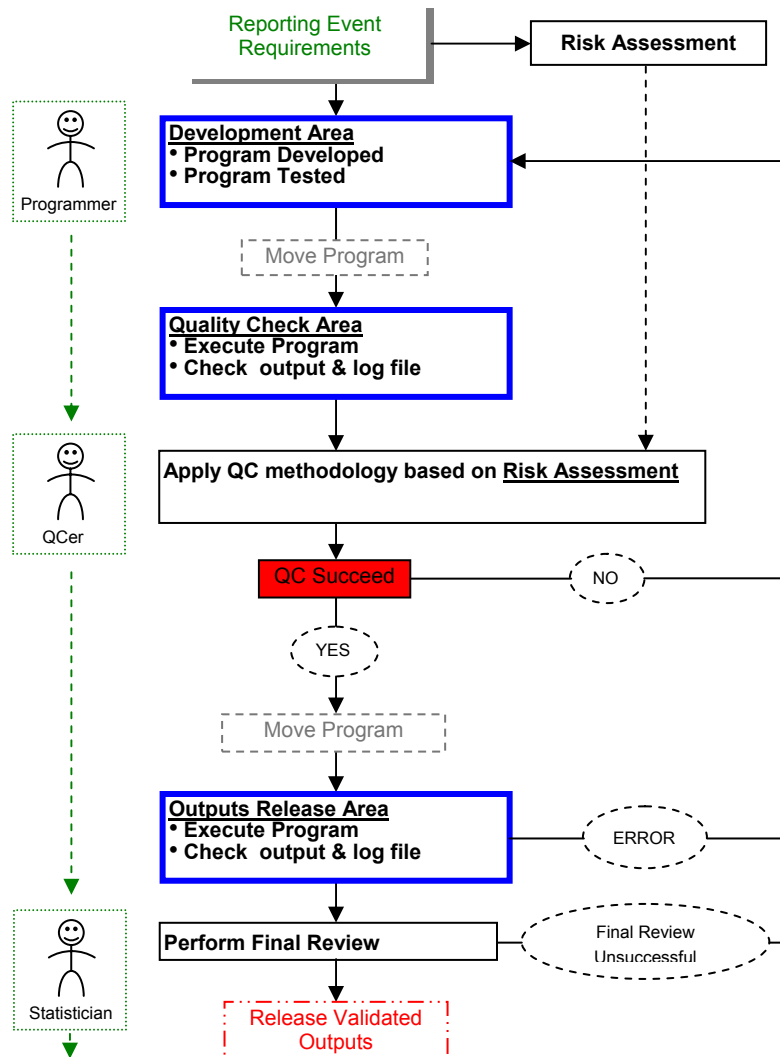


Figure 3 illustrates the process of developing a SAS program

In addition to the assessment of the reporting event risk, the outputs' risk has to be defined. This requires the assessment of all the outputs individually (e.g. VADs, TFL, etc) in order for them to be grouped according to their risk. Similarly with the risk of the Reporting Event, there are a number of factors that contribute towards a decision taken for the risk of each output. These factors include; (i) sensitivity of the data, (ii) the complexity of the SAS code, (iii) the origin of programs that are used to produce the report objects (e.g. central macro library), (iv) the sensitivity to data changes and (v) the likelihood to identify the error. Both the risk of the reporting event and the risk of the outputs need to be recorded in the validation document.

PhUSE 2007

Teams will also need to decide how to treat programs that have been created for the purpose of other reporting events prior to coding. Also, teams need to decide on the data that will be used to carry out the programming, i.e. test data or real dataset. Finally, the programming teams need to plan and record the approach to be used for testing the SAS programs and outputs.

2. Program Development

At this stage, the programmer needs to be fully familiar with the user requirements and follow all the programming conventions the team has decided during the planning stage, e.g. naming conventions, etc. The developer needs to first assign the risk for the program, which should be at least as high as the highest risk of the outputs that the program creates (Figure 3).

During this step the programmer may change the risk of the output if there is strong evidence to do so. All the necessary information must be included in the code in order to enhance traceability of both the developer and the outputs. It is essential that the program meets the user requirements and the programmer is responsible for that. Before releasing the program for testing (Formal Testing), the developer needs to ensure that the desired output has been created. This includes some testing that the developer will carry out, e.g. usage of defensive coding.

3. Quality Check (QC)

The Quality Check (also known as the Formal Testing) is performed by a different programmer who must ensure that all elements of *Plan and Program Development stage* have been taken into consideration to generate the program and all the requirements are met.

This is an essential step as the time that a programmer spends on testing the code depends on the risk of the program, i.e. high, medium and low. A high risk program needs to consider using more testing approaches to reduce the possibility of errors present. On the other hand, a low risk program means that the possibility of the results being misleading is limited and the programmer needs to spend less time validating the code. This reduces the time of the overall validation process as the programmer spends more time checking high risk programs and less time on any medium or low risk programs.

A number of different testing procedures are deployed and it is up to the programming team to decide the combination of those to be applied. Whichever combination the team decides to use, the programmer should consider the following:

- Ensure the outputs to be created by each program are clear, and follow predefined name and location of both the program and output, respectively.
- The user requirements are clear and the data to be used is well known
- The SAS code is robust and contains adequate comments
- The log files are free off errors, warning and other unacceptable messages
- The programs are executed on the latest data and the output files match the defined requirements

The QC methodologies include;

3.1 Program Review

For this method, the programmer focuses on reviewing the code line by line to mentally picture the data manipulation in order to achieve the outputs defined by the user requirements. In addition, the programmer should check if the code is robust, readable, follows a logical order that helps people to understand it and contains useful comments. The location and the name of the output should also be verified, so that it matches what the team decided at the planning stage. Program Review is recommended to be used for programs with minimal data manipulation, limited data combination and programs that generate listings.

3.2 Output Review against input data

The person following this method needs to verify that the values displayed on the outputs reflect those of the input data or simple calculations are displayed correctly on the output. In addition, the programmer confirms that the output is well presented. This approach is suitable for checking listings, graphs, and summaries of demographic type data.

3.3 Output Review against requirement document

Similarly as the Output Review against input data approach, this method aims to confirm whether each output matches the user requirements, e.g. the number of columns, the column labels, titles, footnotes, the order the

PhUSE 2007

summary statistics are displayed, the number of decimal places, etc. As before, this method should be used to check listings, summaries and graphs.

3.4 Cross checking of outputs against other outputs

This method requests the programmer to check the results displayed on one output against another that contains the same information. The outputs checked may not necessarily display the same format. For example, a summary or a listing output can be checked against a graph that displays the same data.

3.5 Double Programming

This is a common method that is used for all the high risk programs. The same output is generated twice by two independently written codes (the original program and the testing program) based on the same user requirements. The outputs are finally compared to identify inequalities in the derivation and displaying of the values. The tester also verifies the correct use of the variable names, variable attributes, decimal places of the values displayed, etc.

3.6 Amalgamation of methods

This is the most comprehensive form of a testing method. It is a combination of all the methods described above.

In addition, there are other QC approaches not as widely used, within Roche, as the ones described above, i.e. Unit Testing.

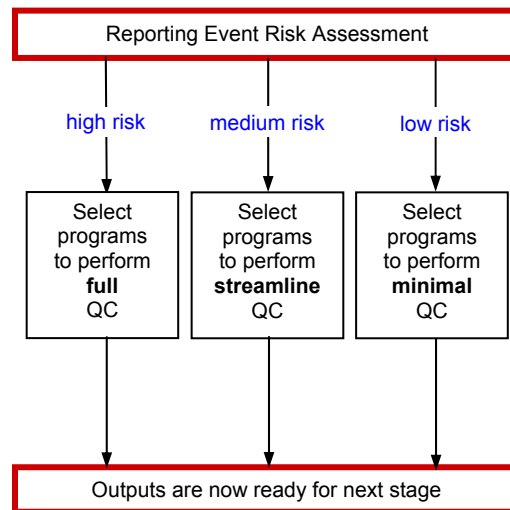


Figure 4 Illustrates the QC methodology based the risk based approach

4. Final Review Check

The Final Review Check intends to endorse the outputs by checking for potential problems that may arise. As mentioned at the planning stage of the process, the Final Review Check method is associated with the risk assessment of the Reporting Event.

If the risk of the reporting event is high, then a full *check* is required. This means that a detailed list of items needs to be created, which describes all the checks against the outputs. The statistician performing the checks needs to supply a written acknowledgement to confirm the successful execution of this list.

If the risk associated with the report object is medium then a streamlined Final Review Check needs to be performed. For such purpose, a checklist (a less detailed list of items that need to be checked against the outputs) is created and followed for the chosen outputs. The checklist needs to be signed by the person executing it.

In cases where the risk assigned to the output is low, then the Final Review check is minimal. In fact no formal checking is required, just a written acknowledgement from the statistician confirming that the outputs are acceptable.

All final Review check documents need to be stored and referenced in the validation document.

PhUSE 2007

If an output fails the Final Review check it is either because of a fault in the program, in which case the programmer is notified in order to review the code, or the user requirements aren't met, in which case the user requirements are reassessed and the programmer needs to repeat any necessary work.

5. Release & Use Outputs

After successful completion of the Final Review all the outputs can be used to create the Clinical Report Study. In case further changes are required, the team needs to either start a new validation process (with a new Reporting Event) or update the current validation process (with the existing Reporting Event).

CONCLUSION

Patients' safety is very important and needs to be considered seriously by everyone. It is understandable that a lot of terms the regulatory authorities use are interpreted differently across the industry due to the fact that such organizations attempt to broadly word their regulations. However, the message is that all the SAS programs must be accurate on their purpose and it is the regulatory authorities' responsibility to manage that this is happening.

On the other hand, health industries need to prove to the regulatory authorities that their work is compliant with all regulations. In specific, biostatistics teams need to prove that there are not errors in the SAS programs and show control over them to be able to prove that the programs work correctly. Failure to do so can result in delays in submission, credibility problems and may also have an effect on public health. The validation method that uses a risk-based approach reduces the risk of such unwanted occasions to occur. The approach described in this paper has been used and its effectiveness and values are visible in our process.

REFERENCES

1. Guidance for Industry, Part 11, Electronic Records; Electronic Signatures – Scope and Application, Food and Drug Administration. August 2003
2. General Principles of Software Validation; Final Guidance for Industry and FDA Staff. Food and Drug Administration, January 2002.
3. Pharmaceutical cGDMs Initiative, Deputy Commissioners, Food and Drug Administration
4. Risk Based Approach to SAS Program Validation, Keith C. Benze

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Nikos Tsokanas
Roche Products Ltd
6 Falcon Way
Shire Park
Welwyn Garden City
AL7 1TW
Tel: +44 1707 36 5889
Email: nikos.tsokanas@roche.com