Paper FC04

Risk-Based Approach to SAS® Program Validation

Keith C. Benze
SEC Associates, Inc.
3900 Paramount Parkway, Suite 150 South
Morrisville, NC 27560

ABSTRACT
SAS is widely used throughout the FDA regulated industry, even by FDA itself; however, the FDA regulations for the validation of computer systems vary across the regulated areas, and it can be difficult to understand if and how these regulations apply to SAS programs. In order to address this issue, it is important to understand the basic expectations for FDA and industry for the integrity of data, and to design a process for the use and verification of SAS programs based on these expectations and how SAS is used. This paper will discuss the regulations, along with FDA and industry expectations, and present a risk-based approach for determining what level of validation and documentation is appropriate for SAS programs.

INTRODUCTION
SAS® is widely used throughout the pharmaceutical industry for many different tasks. The most common usage is in the analysis of data from non-clinical and clinical studies; however, it is also widely used in many areas of pharmaceutical development and manufacturing for a variety of purposes. The FDA regulations, in terms of being specific about computer compliance (i.e., validation and control) vary across the different regulations. In addition, service providers (e.g., contract research organizations, contract laboratories, etc.) have to comply with customer expectations, which can be even more stringent than what is expected by FDA. Even though expectations vary, there is one consistent message being sent by FDA and industry; *data and records being produced and maintained by computer systems must be complete and accurate, and the data integrity is the responsibility of the user or owner of the data*. If you are using SAS programs to generate data for any purpose that falls into a regulated area, the results of those programs are expected to be correct, and some level of verification or validation maybe expected.

Before you decide that this paper is a waste of time because there is no need and no resources to validate SAS programs, it is important to understand that validation is nothing more than making sure that your program does what it is supposed to do. Any good programmer does this instinctively. Is it really unreasonable to expect that programs work correctly and that the data is complete and accurate, particularly when the programs are used to determine if drugs are safe and effective? The purpose of this paper is to help identify when validation is appropriate and ways to approach validation that will add value to your programs, while not requiring inappropriate amounts of work. If the word “validation” is offensive to you, please substitute words like “verification” or “quality assurance” or anything that works for you. The word “validation” is being used here in its true meaning and will be explained in more detail later in the paper.

TERMINOLOGY
For the purposes of this paper “SAS” will refer to the “base SAS” software and “SAS programs” will refer to the programs developed using base SAS (and possibly other SAS tools). Consequently, when this paper discusses the verification or validation of a SAS program, it is referring to the program written using base SAS, and not the base SAS software itself. In other words, you are validating or verifying your use of the SAS software, not the base SAS software.

The end user of a SAS program may not be someone who is actually running the program code, as is typical with other types of computer systems. The end user may be someone who is using the data and/or analysis that is produced by the SAS code, and may never have any direct interaction with the program. Therefore, for the purposes of this paper, the end user may be either someone who executes the SAS code or someone who uses the output of the program.

In the FDA regulatory environment, the term “predicate rule” is often used and can cause some confusion. The predicate rules are the existing regulations and/or laws in support of the Federal Food, Drug, and Cosmetic Act and Public Health Service Act (e.g., 21 CFR Parts 58, 211, and 820), with the exception of 21 CFR Part 11 (the electronic records; electronic signature rule). Part 11 states that any computer system that is used to satisfy any regulatory requirement for recordkeeping or signatures must comply with the requirements of Part 11. Consequently, part 11 is based on, or predicated by, all of the other regulations; which are commonly referred to as the predicate rules.

FDA CONCERNS
It is important to understand that the purpose of FDA is to protect the public health. With pharmaceuticals, this means assuring that the drugs supplied to the public are safe, effective, and high quality. In order to accomplish this task, FDA has

instituted regulations to define the processes required for the approval and manufacturing of drugs. The only way that FDA can assure that the regulations are being complied with is through the examination of records and the evaluation of data. Consequently, records that are generated by computer systems are of particular concern.

The scope and application guidance for part 11[1] states that you must comply with all applicable predicate rules for validation. If there are no regulatory requirements to validate a computer system, then the decision whether or not to validate the system should be based on the impact that the system has on your ability to meet the predicate rules. In addition, the guidance recommends that the approach to validation be based on a justified and documented risk assessment and the potential of the system to affect product quality and safety, and record integrity. This clearly expresses the expectation that computer systems should be validated; however, is this same expectation applied to the SAS programs used to satisfy regulatory requirements? To help answer this question, FDA experts in each of the regulated areas for pharmaceuticals (non-clinical, clinical, and manufacturing) were contacted and asked the following questions:

- Is FDA concerned about the validation of SAS programs that are used to analyze data in pre-clinical, clinical, and manufacturing environments? This is not referring to validating the SAS program itself (i.e., base SAS), but the actual SAS programs that are written for data analysis.
- During FDA inspections, would an investigator ever ask about the validation of SAS programs? This would include if programs are validated as well as the procedures used for validation.

The general message from the FDA experts across all areas is that the data being generated by SAS programs is expected to be complete and accurate, regardless of the regulatory requirement to validate a system; however, there were some differences in the expectations for the validation of SAS programs. Please note that one issue with FDA providing direct information is that it is often taken literally and becomes "policy", rather than being considered as the general advice it is intended to be. Consequently, the information gathered from these officials is summarized below and not quoted specifically to the source.

NON-CLINICAL STUDIES

The regulations for good laboratory practices for non-clinical studies (21 CFR 58 - GLP) contain a specific requirement for the validation of computer systems used to satisfy the regulatory requirements (§58.63). This section of the regulation would require that any SAS programs used to analyze data required by that regulation would need to be validated. FDA confirmed this, but admitted that the validation of a SAS programs is not as much of a concern as the validation of the primary data collection systems. If there is an issue with the analysis, FDA can request the raw data and reanalyze the data; hence, the emphasis on the primary data systems. However, if FDA reviewers reanalyze a study in an application and find an error, they will either reanalyze everything or send it back to the sponsor to resolve the issues. Either way it causes a delay in the review of the application and brings into question the credibility of the sponsor.

CLINICAL TRIALS

Although there are no specific requirements in the FDA clinical trial regulations (GCP) for the validation of computer systems, FDA has said that it is "not unreasonable" for FDA to be concerned about the validation of SAS programs. When FDA reviews a submission, the primary goal is to verify that the data and data analysis submitted support the claims made by the sponsor. If FDA has a question about any of the data, they may request more data and reanalyze. If the reanalysis of data and a review of procedures and protocols do not resolve the issues, FDA may dig a little deeper into the procedures used for the development and validation of the SAS programs. In one example cited by FDA, there were adverse events reported that were not included in the statistics. Upon review of the SAS programs, it was determined that there was an error in the SAS program that was truncating the data and causing the omission. As with non-clinical data, this process will cause significant delays in approval and bring into question the creditability of the sponsor.

MANUFACTURING

SAS programs can be used for a variety of purposes in a manufacturing environment (e.g., annual product reviews, stability studies, calculating test results, process validation protocols, etc.). The regulations for Good Manufacturing Practices (GMP) have a specific requirement for the validation of computer systems used to satisfy the regulatory requirements (§211.68). There is a general expectation in manufacturing environments that any software used to satisfy a regulatory requirement is validated, including SAS programs. Depending on the use of the SAS program, it is likely that FDA would expect to see the SAS programs validated and controlled.

GUIDANCE DOCUMENTS

There are guidance documents issued by FDA that discuss the need for validation, including the guidance for the scope and application of part 11. Although these guidance documents are not legally binding on either FDA or the industry, they do present FDA's current thinking on the topic. In the guidance on the use of computerized systems in clinical trials[2] it states that "FDA may inspect documentation, possessed by a regulated company, that demonstrates validation of software." In 2004, FDA issued a draft for the revision of this guidance[3] in which the wording about validation of software has been revised to be consistent with the part 11 scope and application guidance; "Even if there is no predicate rule requirement to validate a system, it may still be important to validate the system, based on criticality and risk, to ensure the accuracy, reliability, integrity, availability and authenticity of required records and signatures." Regardless of the wording, the expectation is that the software used must work correctly.

In addition to the FDA guidance documents, there are also international guidance documents issued by organizations such as the International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use (ICH) and the Pharmaceutical Inspection Convention; Pharmaceutical Inspection Co-operation Scheme (PIC/S). As with FDA guidance, these documents are not legally binding in the U. S. but are used by other countries as standards and should be reviewed if doing business internationally.

INDUSTRY EXPECTATIONS

The sponsor of the data submitted to FDA is ultimately responsible for the completeness and accuracy of the data and for complying with all regulatory requirements. If a sponsor contracts with a service provider (e.g., contract research organization, contract laboratory, or contract manufacturer), the sponsor is still responsible for the work being conducted by the contract organization. For example, a warning letter issued by FDA to Cypress Bioscience, Inc.[4] stated, "Cypress failed to have a system in place to verify the accuracy of data collected at laboratories not under their direct control. Transfer of data sets, including laboratory data from the contract laboratories to the CRO, to your firm and then transferred again to the contracted statistician resulted in discrepancies in what should have been identical data sets." As with FDA, the sponsor's primary concern is that the data is complete and accurate, but sponsors may expect service providers to have formal procedures and documentation for the validation and control of SAS programs.

VALIDATION

Having established that there are both regulatory and business justifications to assure that programs work correctly (i.e., validate programs), it is necessary to understand the basic concepts of validation so that techniques for scaling the validation effort can be discussed. The word "validation" has a bad reputation in the regulated industries and unfortunately, some of it is well deserved. When the word "validation" is used, everyone immediately thinks that there will be a ton of documentation required which will increase the need for resources, the overall cost will sky-rocket, and it will take forever to complete the project. And the worst part of this scenario is that most people believe that all this work is not necessary and that it does not add any value; it is only an exercise in futility required to satisfy the FDA. *But it does not have to be that way!* The truth is that validation is really nothing more than making sure the program works correctly, and it can be argued that making certain that the program works correctly really does add value. Contrary to popular belief, FDA is not in the business of making life more difficult for the industry just for its own entertainment. FDA is responsible for protecting the public health by assuring that products available to consumers are safe and effective. FDA started discussing validation over 20 years ago when computers were first introduced into regulated processes because they understood the importance of computers in terms of protecting public health. Their primary concern is that all data, whether generated by computer or not, is complete and accurate, but there are specific concerns that must be addressed when the data is generated by a computer. Consequently, validation should be done to assure that data is complete and accurate, and not for the sheer joy of going through the process. Let the purpose of validation drive the effort; *focus on the principles, not the process*.

Try for a moment to forget all of the validation terminology that you have heard or been taught and think of validation as three activities:

- *Specification* - determining what the program is supposed to do
- *Verification* – making certain that the program does what it is supposed to do
- *Control* - maintaining control over the program to assure that it consistently works as expected

The key to an efficient validation process is to satisfy these three activities using the appropriate amount of resources based on how the system is used as part of the overall process. This is the basis for a risk-based approach. FDA expects organizations to understand how computer systems are used as part of their overall process and to do the work that is necessary to assure that the system works correctly based on the effect the system has on the final product (which may be a submission to FDA). As one FDA official put it, "Good engineering and good processes meet regulatory intent. If it does not, then there is a real problem." Notice that there is no mention here of specific terminology (IQ, OQ, PQ, etc.), but the focus is on the principles or activities required to assure that a computer system functions correctly.

Once you understand the real purpose of validation, then you will also understand that it is not possible to partially validate a system. Any validation effort must, at a minimum, include specification, verification, and control activities. A well defined, efficient validation process will assure that these activities are completed, while providing an approach that is scalable based on how the system is used and the effect that the system has on the final outcome. The word "scalable" is used to describe a process that addresses all three of the key activities (specification, verification, and control), but allows for a different level of documentation and/or effort based on certain factors like program risk, complexity, use, etc. It is important to differentiate between the words "scalable" and "flexible". Sam Clark, an ex-FDA computer expert, refers to the term "flexible procedures" as an oxymoron. The purpose of procedures is to define a process that can be followed to consistently produce an end result. If the procedure is "flexible" then it often leaves the process open to interpretation causing inconsistency and defeating the purpose of having a procedure. A "scalable" process is one that provides for consistency and therefore produces a consistent end result, but allows the overall effort required to produce the end result to be scaled based on certain factors.

One argument against the need to validate SAS programs is that SAS has been around for years and is used by everyone, including FDA. Certainly SAS is a quality product and has a long established track record in the industry, but like all software, SAS has its limitations. Validation is important because it proves that the program does what you want it to do, in the way in which you are going to use it. In other words, SAS may be 100% correct, but if used incorrectly it can produce an unexpected

result. Incorrect does not necessarily mean an error in the code or that a procedure was wrong; it may mean that the output of the program is not what the end user needs. Validation assures that the end user needs (requirements) are clearly stated and that the program meets those requirements correctly and consistently. As a point of caution, the fact that FDA uses SAS extensively may actually be part of the justification for validating SAS programs. FDA has resources with extensive knowledge and experience in using SAS, which means that FDA knows what to look for and where to look.

RISK-BASED APPROACH

The scope and application guidance for part 11 uses the phrase "documented and justified risk assessment", but how is this applied to the validation of SAS programs? The dictionary definition of risk is the possibility of loss or injury. If you apply this definition to a program being used to satisfy a regulatory requirement, then risk is the possibility that the data generated will not be complete and/or accurate, which will result in some negative effect. The best approach to reduce or remediate the risk, is to assure that the program will work correctly and consistently, or in other words, to validate the program. The amount of effort that is spent on the validation should be in proportion to the outcome if the program does not work as expected.

There are a number of standards in the industry for conducting risk assessments. Unfortunately, most of these techniques are often too complicated, time consuming and not appropriate for SAS programs. However, if you focus on FDA's primary concerns and the basic principles of validation, a fairly simple, risk-based approach can be developed for use with SAS programs.

The first step in a risk-based model is to determine which programs should be validated. In order to make this determination, you must understand how the programs are being used as part of the overall process. This requires that you understand the process (data analysis for a submission, annual product reviews of the quality in a manufacturing process, etc.) as well as how the programs will be used as part of that process. The key here is to determine the effect on the process if the program does not produce the desired results. For example, if you are analyzing safety data and the analysis is not correct, could this put the subjects of a trial at risk of injury or death? A program that has a direct effect on public health, or is generating statistics that are critical to proving that a drug is safe and effective, should be considered as a high risk, and some level of validation should be completed. Also, based on the part 11 scope and application guidance, if there is a predicate rule requiring that the program be validated, the program should be considered high risk, and some level of validation should be completed. If the program will not have a direct effect on public health and there are no regulations requiring that the program be validated, then guidance documents or international standards should be reviewed to determine if there are any requirements or recommendations for validation. Remember that these documents may not be binding on your organization, so the decision to validate a program based on guidance documents is more of a business decision based on the risk threshold of your organization and whether these documents are binding in any of the markets where you are doing business. Any programs that do not fall into either of the previous two categories can be considered low risk and validation is likely not an issue. Note that there are essentially two types of risk being assessed; risk to public health and regulatory risk. Risk to public health should never be taken lightly, and programs that could potentially affect the safety of the public should always be validated.

Once it has been determined which programs should be validated, the next step of determining the level of work necessary to validate the program, is not so cut-and-dry. There are several factors that should be considered when determining what level of documentation and testing is required, for example:

- Program use – multiple use vs. single use – in general, multiple use applications are higher risk because the source of the data is often unknown
- What is the composition of the input data – could there be aberrant or missing data
- How will the type of input affect the output – will any aberrant or missing data generate errors and if errors are generated, how will the program handle these errors
- Complexity of the code – compound text or dataset manipulation, long calculations in a data step, iterative loops, logic statements, etc.

The actual factors that must be considered will vary depending on your organization and the use of SAS programs.

SPECIFICATION

The requirements for the program should be clearly stated and testable so that the output of the program can be verified to meet these requirements. The requirements should include the input data, any processing and/or calculations, and the output. If a document already exists that specifies the requirements (study protocol, statistical analysis plan, etc.) the existing documentation may be used and a separate requirements document may not be necessary.

The design of the program, or how the program will be structured to meet the specifications, should be also be documented. Depending on the risk and complexity of the program, the design may documented as part of the code or it may be a separate document. For example, if the program is a single-use program with a known input, the design can be documented in the actual code following approved coding standards. However, if the program is intended to be a multiple use program that will be executed using unknown inputs, the program is likely to be more complicated and require more planning and design; consequently, a separate design document may be more appropriate. The requirements should be traced to the design so it can be easily determined which elements of the design are being used to satisfy which requirements.

The requirements and design should be approved prior to writing the actual code to assure that the end user and programmer both understand what the program is intended to produce. This sounds like a lengthy process, but for a short, simple program it can be accomplished very quickly and efficiently. For a high risk, complex program, it is appropriate to spend a little more time up front to make sure the program is designed correctly. Keep in mind that even if the end user is a statistician, they may not be a SAS programmer, and a SAS programmer may not necessarily understand what the end user is trying to accomplish. Consequently, if the end-user and the programmer are not the same person, it is very important that both understand what is to be accomplished and how it will be done. Defining and agreeing to these specifications prior to writing code saves time and effort, and reduces the risk of errors. Software development metrics show that finding and correcting problems during the specification phase of a project can cost 40 – 50 times less than when errors are discovered once the code is written and the program put into use.

VERIFICATION

The level of verification or testing can be determined based on the requirements and the factors listed above. This might be as simple as a peer review, or as complicated as designing a series of test cases with known input data and expected output that will verify that the program executes correctly. Based on the factors of risk, complexity, etc. a test plan should be documented that details the approach that will be used and the justification for that approach. At a minimum the code should be reviewed by a qualified SAS programmer other than the programmer who wrote the code, and the output log file should be reviewed for warnings and errors, and all errors should be resolved. For more complicated programs, particularly multiple-use programs where the input data may be unknown at the time of development, test cases should be written which use known input values to verify that the program will correctly handle all ranges of expected data, including aberrant data and missing values. Test cases should be traceable back to the requirements to demonstrate that all requirements have been met. All testing should be documented, signed and dated by the tester, and reviewed, signed, and dated by a second party stating that they agree with the conclusion drawn by the tester. After completion of testing, a short report should be written and approved that summarizes the results of the testing and determines if the program is fit for use.

A coding standard for SAS programs should be documented and approved. The standard should include both programming standards and naming conventions to be used with developing SAS programs and should be used when executing all code reviews. Using a consistent standard will result in programs that are well documented and easily maintained. Following are some examples of what should be included in the programming standard:

- The information that should be included in the program header (e.g., program name, who created the program, the version of the program, the version of SAS used to create the program, the purpose of the program, a history of change, etc.)
- Setup module – the SAS procedures and statements that are necessary for setting the environment for proper execution (e.g., clearing temporary datasets, clearing variables, etc.)
- Program comments
- Naming conventions

It is important to maintain a level of objectivity in writing, testing, and reviewing the code and test cases. The objectivity is necessary to catch errors and should not be viewed as a trust issue with any of the personnel. Writing SAS programs can be very complicated and it is always good to have a second pair of eyes review code and documentation. Most organizations do not have the luxury of unlimited resources with the qualifications and training to have separate people write code, write test cases, test code, and review results. It is possible to have only two or three people involved in the process if it is designed to assure some level of objectivity. For example, it may be possible to have one person write the code, write the test cases, and execute the test cases, while having a second person review the code and the result of the testing. It is important, however, that anyone involved in the writing and reviewing of SAS programs have sufficient qualifications and training to understand what they are reviewing and be able to identify issues and errors.

Independent verification of off-the-shelf SAS functions and procedures may not be necessary depending on the risk of the program and how the functions are used. Keep in mind that some of these functions may have limitations. You should know these limitations and particularly if you are exceeding the limitations, the testing should be much more robust. Some organizations will identify the critical procedures that will be used and will do a one-time validation of those procedures. This can be done by using standard datasets with known outputs or using other statistical programs to verify the results. Whether or not this is necessary depends on the risk of your programs and the overall risk tolerance of your organization.

CONTROL

All input data, output data, program code, and documentation must be maintained and controlled. It may be necessary to re-analyze data or submit datasets to FDA for further analysis. Consequently, there should be a process that assures that any changes made to the data, programs, and/or documentation is controlled and approved. This can be accomplished through automated, technical resources or through a combination of technology and procedures. For example, control can be accomplished using tools designed specifically for the purpose of controlling SAS data and programs (e.g., SAS® Drug Development), using configuration management tools (e.g., Microsoft® Visual SourceSafe), or using the operating system access controls. There should be written procedures for the use of any of these tools. The detail in the procedures will depend on the level of control offered by the tool; if the tool offers less technical controls, then the procedure will require more detailed controls. For example, if operating system controls are used to protect files, then the procedures will need to include how the files will be named and where files will be stored. Automated tools may handle these issues automatically.

STANDARD OPERATING PROCEDURES

A series of standard operating procedures (SOP's) should be developed and approved to provide a consistent process for determining which SAS programs need to be validated and the documentation required for the validation. The procedures should provide the necessary information for determining which programs need to be validated and what documentation is required. All of the factors that have been discussed in this paper need to be taken into consideration when designing the SOP's. For example, the SOP's can address the regulated area in which SAS will be used (i.e., GLP, GCP, and/or GLP), the type of programs that will be used (e.g., multiple use vs. single use), the personnel that will be involved in the development and testing of code and the review of documentation, the qualifications and training of personnel, the risk tolerance of your organization, etc. The process needs to be scalable based on these factors to assure that the appropriate level of work will be completed for each program. Following is a list of topics that should be covered by the SOP's:

- Validation life cycle – describes the overall process, the documentation required, and who is responsible for what tasks
- Requirements and design – describes the need for and content of the requirements and design specifications
- Testing – this is sometimes referred to as the quality plan and describes the verification activities required (code review, testing, log review, etc.)
- Program and data management – describes the control of the SAS programs and datasets during and after validation
- Coding standards – describes the techniques that should be used in the development of programs including comments, naming conventions, etc.
- Change control – how to make changes to programs that have been validated, and to assure they continue to function correctly

It may not be necessary to have separate SOP's for each of these topics. Depending on the organization it may be more efficient to combine some of these topics into single procedures.

In addition to the procedures, templates should be created for the required documentation. Templates should be designed to capture the required information while keeping the amount of effort to a minimum. For example, a single form can be used to capture the requirements, design, and the traceability from the requirements to the design. Another form can be used to document the test plan and test summary. Templates not only add efficiency to the process but also assure conformance to the procedures by making certain that all required information is completed. Templates can be an appendix to the procedure so they are reviewed, approved, and controlled as part of the procedure. If templates are maintained as separate documents, they must be controlled to assure they are kept up to date with the procedures.

Designing, writing, and approving procedures may take some effort; however, the time spent can be well worth the effort. Providing a documented process will assure consistency, reduce errors, and build efficiency as everyone will know what is expected and have the tools available to meet those expectations. We have worked with a number of clients to help them develop this type of life cycle and the process has been very beneficial. One client, a contract research organization, was audited by a sponsor, and one of the first questions asked during the audit was to review the life cycle for the validation of SAS programs. Having these procedures in place not only answered all of the sponsor's questions, but raised the confidence level of the sponsor. At least one other client has said that using this process, they have found problems that would have gone otherwise unnoticed.

SUMMARY

The general expectation by FDA and industry is that data generated through the use of SAS programs must be complete and accurate. Errors in SAS programs can cause delays in the review of submissions, creditability problems, and possibly even affect the public health. Consequently, controls should be used to assure that SAS programs work correctly and consistently. The controls should be designed to be scalable based on the use of SAS and the outcome if the program is incorrect or does not meet the stated requirements. This type of risk-based approach will not only assure the integrity of the data, but can also build consistency and efficiency, and add value to the overall process.

ACKNOWLEDGEMENT

I would like to thank Lisa Olson for all of her assistance in the research and preparation of this paper, as well as for her expert help in developing the overall approach presented.

CONTACT INFORMATION

Your comments and questions are encouraged and appreciated. The author can be contacted at:

Keith Benze
SEC Associates, Inc.
3900 Paramount Parkway, Suite 150 South
Morrisville, NC 27560
919-467-1242
kbenze@secassociates.com
www.secassociates.com

REFERENCES

[1] *Guidance for Industry, Part 11, Electronic Records; Electronic Signatures – Scope and Application*. Food and Drug Administration, August 2003.

[2] *Guidance for Industry, Computerized Systems Used in Clinical Trials*. Food and Drug Administration, April 1999.

[3] *Guidance for Industry, Computerized Systems Used in Clinical Trials – Draft Guidance*. Food and Drug Administration, September 2004.

[4]Warning letter issued to Cypress Bioscience, Inc., item 3. Food and Drug Administration, Center for Devices and Radiological Health; June 7, 1999.