Benefits of Validating SAS Systems in a Regulated Environment

Sy Truong, Meta-Xceed, Inc, Fremont, CA

ABSTRACT

Validating a new version of SAS on a production server used to be a daunting task. The SAS System version 9.1.3 ships with user friendly installation qualification tools. This is coupled with existing tools that make it easier to validate SAS. Besides qualifying the installation, there are other tasks and components of the system that need to be validated or verified. Some of these components include:

1. Backward compatibility issues with older versions of datasets and format catalogs.

2. Validating multi use macros and standardized code templates.

3. Verifying stand alone or project specific programming and output.

4. Affects on standard operating procedures and programming practices.

The interconnectedness of the SAS computing environment does require considerable efforts in validating a SAS System. However, if this is executed successfully, it can allow for greater traceability between output, programs and source data. The performance qualification also sheds light on ways of optimizing the work and dataflow of your computing environment. The many benefits of performing validation of the SAS System will outweigh the costs. In addition, it is a requirement within a regulated environment so it is recommended to be prepared.

INTRODUCTION

Validation of a SAS system most commonly occurs during an upgrade from an older version of SAS or moving to a new platform. The examples used in this paper include migrating from SAS 8.2 to SAS 9.1.3 and moving from a legacy operating system to the windows platform. In either case, similar validation challenges are confronted. It is recommended that you first acquire a global view of the system and identify the architecture. Only after gaining this perspective would it be useful to then zoom in on individual components. This allows you to access the scope and interconnected of each component so that your validation efforts are balanced and thorough. Once the architecture is clearly understood, the requirements and functional specifications of each component are documented. These functional specifications then drive the validation testing.
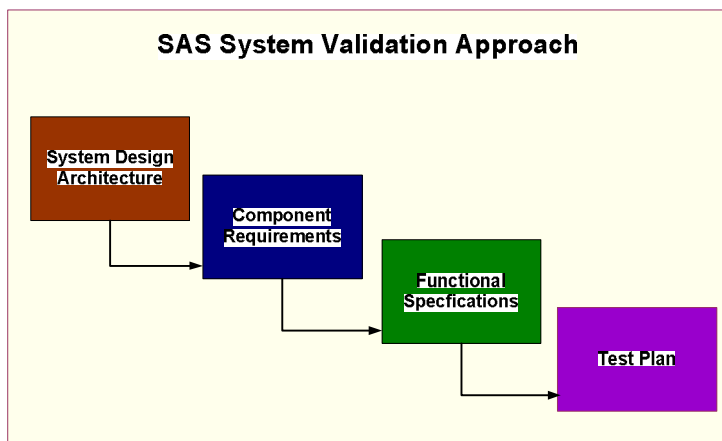
It is important to follow these steps in a systematically and orderly fashion since they are interdependent. Documentation of each step in the validation process is also essential in capturing and proving that the validation effort was done properly. Besides documenting each step, it is also important to capture the traceability of each validation task. For each test case that is

performed, there is an associated functional specification which then is connected to the requirements for a particular component of the system as a whole. The map or traceability matrix that ties all these validation components together is pivotal to an auditor. Proper documentation will make the difference between a successful validation audit and a complete failure.

The main goal of the validation effort is to ensure that the installation and implementation of the SAS system and its associated tools function as intended by the vendor (SAS Institute) and your organization. The validation will ensure this success. In addition to this goal, the documentation of your validation effort will also ensure the integrity of your computing environment and be in compliance with regulatory requirements such as the CFR Part 11 within the biotechnology and pharmaceutical industry.

ARCHITECHING COMPONENTS

The first step in your validation effort is to understand what it is that you are working with. The SAS System as delivered to you in a series of CDs is a system which contains modules such as Base, Stat, Graph and other components of SAS. This however only makes up part of the system that you are implementing in your organization. The SAS software fits into a computing environment that interacts with other software and hardware. If you were to take into account all the associated hardware and software that SAS interacts with, this is what is considered the "SAS System" from a validation perspective. It is therefore important for you to take the right steps to identifying and documenting all these components.

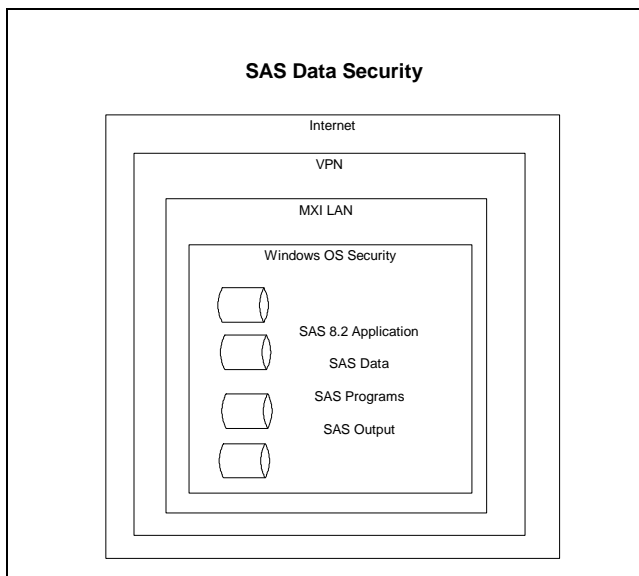**STEP 1:** Identify all the hardware components of your computing environment. For example:

Hardware Component	Name
SAS Application Server	SASAPPSRV
SAS File Server	SASFILESRV
Client Desktops	CLIENTDSK

There could be many physical machines used or just a few. It is important to know where SAS is stored and which machine stores other files that relate to SAS.

STEP 2: Document security model implemented for your SAS environment.

Server Type	Security Levels
SAS Application Server	Windows Login
SAS File Server	Windows Login

There could be a software layer or physical layer to your security model.

SAS Data Security

Internet

VPN

MXI LAN

Windows OS Security

SAS 8.2 Application

SAS Data

SAS Programs

SAS Output

It is common to have a combination of software and hardware contributing to your security model. By documenting it, you will see where there are holes or areas that may need improvements.
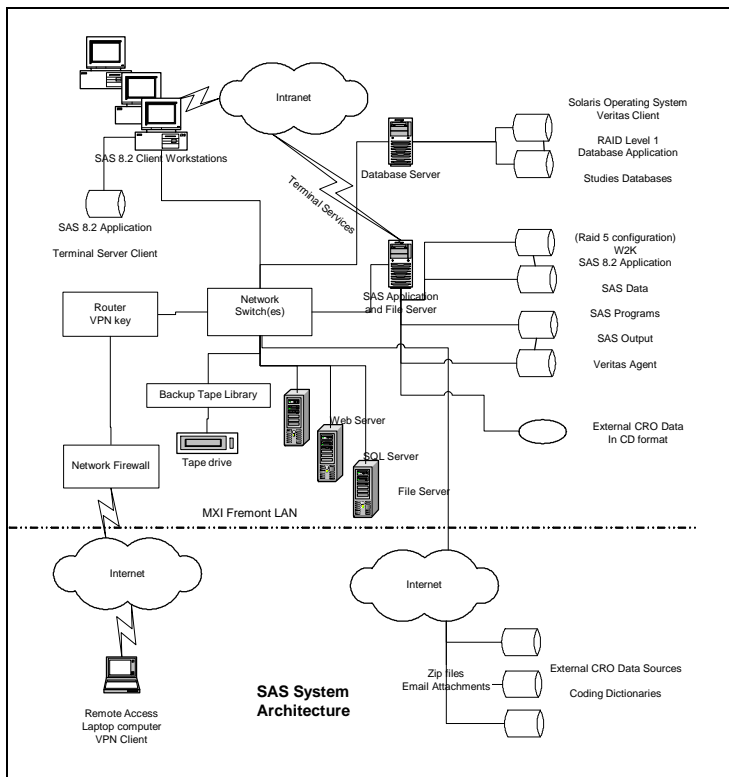
STEP 3: Identify all the software components that are installed on your hardware.

SAS Application Server

- ? SAS 9.1.3 Enterprise Bundle
- ? Microsoft Windows XP
- ? SAS System Viewer
- ? Trialex Clinical File Management Macros Version 2.0
- ? Sy/Validate version 3.0

This step is essential in understanding all the software used on your system. This step helps you perform the impact analysis of how SAS affects other components within your environment.

STEP 4: Document the system architecture. This ties all the hardware and software together.

SAS 8.2 Client Workstations

Intranet

Solaris Operating System
Veritas Client

RAID Level 1
Database Application

Studies Databases

Terminal Services

Database Server

SAS 8.2 Application

Terminal Server Client

(Raid 5 configuration)
W2K
SAS 8.2 Application

SAS Data

Router
VPN key

Network
Switch(es)

SAS Application
and File Server

SAS Programs

SAS Output

Veritas Agent

Backup Tape Library

Web Server

External CRO Data
In CD format

Network Firewall

Tape drive

SQL Server

File Server

MXI Fremont LAN

Internet

Internet

Remote Access
Laptop computer
VPN Client

**SAS System
Architecture**

Zip files
Email Attachments

External CRO Data Sources

Coding Dictionaries

This gives you a bird's eye view of all your components and how they relate to each other. Only include those components that relate to the SAS system. This includes input to the SAS system such as source data bases and output reports or data that is produced by SAS.

SYSTEM REQUIREMENTS

It is nice to have all this technology but you have to know what you want to do with it before it can be useful. If you are the end user, then you would pose the question to yourself.

What do I need SAS for to help me do my job?

See if you can answer this question with one concise sentence. This short answer will be the theme that you will use in capturing your requirements. If you are not the end user, then set up interviews with the end users and ask them the same questions. During this requirement collection stage, distinguish between what the user is requiring and what the software vendor (SAS Institute) is requiring. For example, the system must run on Windows XP Professional rather than Windows XP Home. This may be something that would be nice for the users, but it is more of a vendor requirement. This distinction will help you better organize your requirements.

STEP 1: Identify all the vendor's (SAS Institute) minimum requirements. This has to do with the minimum memory or hard disk requirements for your particular hardware and operating system configuration. This can be different between your SAS sever versus data and client machines.

STEP 2: Document your organization's requirements pertaining to security and computing environments. Your organization may have backup facilities or there are password and data access stipulations. These requirements are usually established by your organization and can be different from the vendor's requirements or the specific user's requirements.

STEP 3: Document the users' wishes and requirements. An example would be that SAS generate reports and perform specified statistical analysis. Some user requirements may relate to step 1 and step 2. For example, the user may want the system to be fast and available 24x7. Sometimes the requirements do crossover. It is not essential that you capture every little requirement but that the most important and essential features are captured.

The requirement steps can be laborious and not very exciting. However, it will give you great insight into what are the core business needs of your organization as it relates to the use of SAS. This understanding is essential in focusing your validation efforts to ensure the highest degree of your system's effectiveness and integrity.

FUNCTIONAL SPECIFICATIONS

Now that you know what it is that you need SAS to do, the next step is to understand how that is accomplished. For example, if one of your requirements is for SAS to generate reports in PDF format, the functional specification corresponding to this requirement would detail the use of PROC REPORT and associated ODS options to produce the required report. The SAS System is very extensive and delivers most of user's requirements. However, there are instances where users have requirements that go beyond what is available from SAS. Here are a couple of sample requirements that may not be delivered by Base SAS.

1. **SAS Program Change Control** – SAS programs that are edited and updated need to be versioned to form a complete audit trail.

2. **Report Table of Contents** – All SAS listings and table output need to be documented and organized in a table of contents.

In this example, the functional specification will detail the use of tools supplied by external vendors such as MXI's Sy/Validate and Trialex tools. In this example the corresponding functional specification would be:

1. Sy/Validate is used to make a backup copy of each SAS program each time it was submitted. This includes recording the user name and time in which it was captured. Optionally, the tool will capture a user note explaining the changes. Any SAS program can be rolled back to an older version. Reports can also be generated to detail a complete audit trail for a comprehensive change control of SAS programs.

2. The Trialex tool is used to organize all report titles and associated footnotes in a central database. A table of contents can be automatically generated in RTF, HTML and PDF format including hyperlinks to each specified report.

This example illustrates that the functional specification provides a little more detail on how a particular requirement is accomplished. If you are the end user, the documentation of the functional specification will answer the question whether the requirement is met. If you are not the end user, you would need to present the functional specification to the user and ensure that it fulfils the original requested requirements before proceeding to the next step.

TEST PLAN PROTOCOL

The last major step in your validation process is creating the test plan and executing the tests. This will confirm that each functional specification that you have identified truly does work. There are many formats in which you can organize your test plan. The following is a suggested step that needs to be documented.

STEP 1: Identify the role and who will be responsible for performing testing related tasks.

QUALIFICATION AND TESTING ACTIVITY	RESPONSIBILITY ASSIGNMENT
Protocol Generation	Validation Specialist or Contractor
Protocol Approval	System Owner Validation Quality Assurance
Protocol Execution	System Owner or Contractor or System Administrator
Protocol Execution Review	Validation
Validation Protocol Deviation Review and Approval	Validation Quality Assurance
Qualification Final Report Generation	Validation Specialist or Contractor

QUALIFICATION AND TESTING ACTIVITY	RESPONSIBILITY ASSIGNMENT
Qualification Final Report Approval	System Owner Validation Quality Assurance

It is not required that specific names be placed but rather the job title or description will suffice. This is more flexible in that the same person can do more than one job or there may be reassignment once the testing starts.

STEP 2: Document the implementation plan. This is where you specify the category of tasks that need to be performed in order for the testing to be completed successfully. An example of some of the areas include:
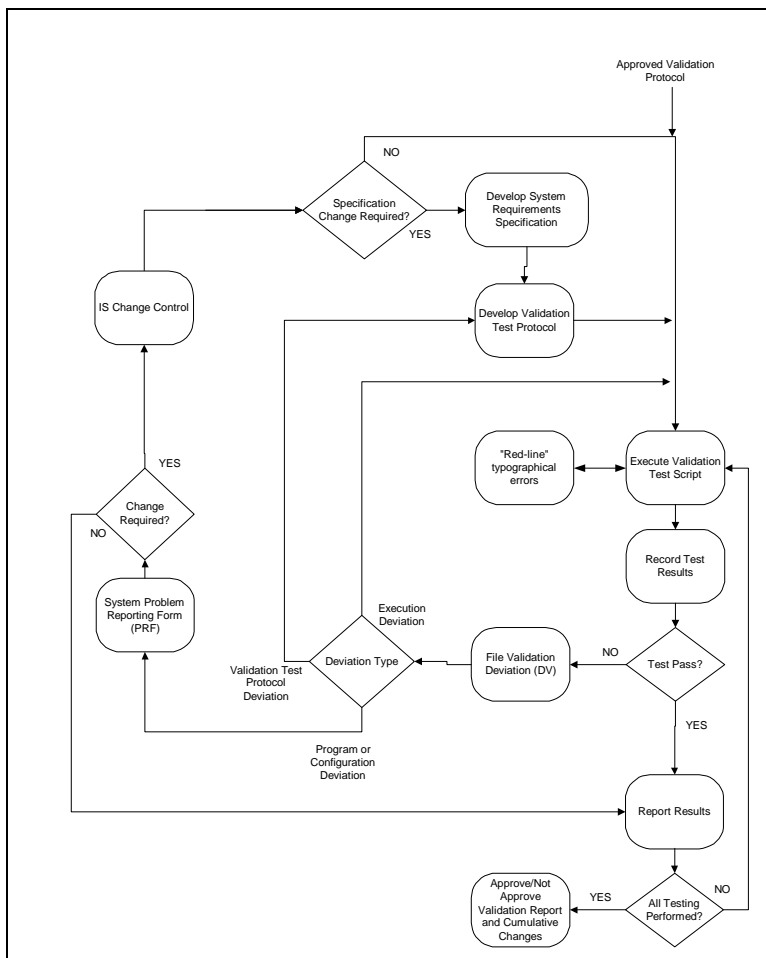
1. **Validation Objective** - The objective of this qualification effort is to qualify the installation and functionality of the SAS system.

2. **Activities**

   a. Installation Qualification Testing

      i. Verification of the hardware and software components

      ii. Verification that the servers are properly secured

   b. Performance Qualification Test

      i. Verification procedures are put in place for the operation of the system

      ii. Backup and recovery procedures are verified

   c. Acceptance Criteria

      i. Demonstrate and document that system requirements have been satisfied

      ii. The system is installed according to the approved qualification protocols

3. Change Control

   a. Version management and configuration management

   b. Revalidation procedures and associated triggers for revalidation

These are just highlighted areas that are to be considered. There is no exact template but these steps are a good start as suggested guidelines.

STEP 3: Define the types of testing that need to be done. There are several different types of qualification testing. You would need to perform all of these to be thorough.

1. **Installation Qualification (IQ)** – This ensures that the installation of SAS and related tools was done according to the vendor's expectation. It usually verifies if all the files and components have been installed and are operational.

2. **Operational Qualification (OQ)** – This ensures that the system is operating with all of the intended features. This includes all the functions detailed in the functional specifications.

3. **Performance Qualification (PQ)** – This ensures that the system performs within the size data and amount of users that would be used in the real world. This will simulate real use to see if the system performs according to expectations.

STEP 4: Testing flow charts can really clarify how testing is to be conducted. This documents the order of the testing steps and how the test is to be handled in the event of deviations.

Approved Validation Protocol

NO — Specification Change Required? — YES

Develop System Requirements Specification

IS Change Control

Develop Validation Test Protocol

YES — Change Required? — NO

System Problem Reporting Form (PRF)

Validation Test Protocol Deviation

Execution Deviation

Deviation Type

Program or Configuration Deviation

"Red-line" typographical errors

Execute Validation Test Script

Record Test Results

File Validation Deviation (DV) — NO — Test Pass? — YES

Report Results

Approve/Not Approve Validation Report and Cumulative Changes — YES — All Testing Performed? — NO

Use diagrams wherever possible to help clarify how testing is to be conducted. This is more useful than any long list of text procedures.

CONCLUSION

During the processing of performing these validation tasks, you may not appreciate what impact it is having. It may seem dry and laborious. In practice, however, you will find that when the test scripts are being created according to the test plan, you will discover issues pertaining to backward compatibility of SAS datasets and catalogs if you are upgrading. You will also find subtle errors and, in some cases, major bugs within your commonly used multi use macros. The performance qualification can also encompass the use of stand alone study specific code. In many instances, this reveals discrepancies that can lead to improvements to your system. The validation process can also solidify and help more clearly define the standard operating procedures and good programming practices of your group. If you can step out of the thinking of having to perform validation grudgingly and see it from a larger perspective, it can have a profound affect on your entire organization in how you and your users interact with SAS and its related data to fulfill your business objectives.

REFERENCES

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Sy/Validate and the Trialex Tools and all other **MXI** (Meta-Xceed, Inc.) product names are registered trademarks of Meta-Xceed, Inc. in the USA.

Other brand and product names are registered trademarks or trademarks of their respective companies.

ABOUT THE AUTHOR

Sy Truong is a President of MXI (Meta-Xceed, Inc.) They may be contacted at:

Sy Truong

48501 Warm Springs Blvd. Ste 117

Fremont, CA 94539

(510) 226-1209

sy.truong@meta-x.com