PharmaSUG 2022 - Paper SI-014

Quality Control with External Code

William Coar, Cytel

ABSTRACT

In the CRO industry, we can create statistical reports in multiple ways: by writing our own source code or receiving source code from sponsors or other external CROs. In projects where we receive source code, the sponsor will provide us SAS code written and verified on their systems. When there is a need for unblinding prior to the final analysis, such as in supporting a Data Monitor Committee (DMC), some sponsors seek an independent Statistical Data Analysis Center (SDAC) to perform this task. The SDAC is expected to run the sponsor provided code in their own environment to produce an unblinded report used for interim monitoring of safety and efficacy.

The primary assumption made about the SAS code is that it is tested/verified/validated (pick your favorite word) by the client. In fact, the SDAC has little control, or sometimes no control, over the SAS code that is received. Even though it is expected that SAS code meets the sponsor's SOPs associated with quality control (QC), additional measures of QC are essential since the code was tested in a blinded manner on a different system. We identify two critical areas where QC is imperative. First is to ensure all programs run on the SDAC system as expected given the potential for different operating systems, SAS licenses and folder structures. Second is to ensure the unblinding process has been verified by the SDAC.

In this presentation we focus on these two areas of QC which we feel are critical when supporting DMCs where source SAS code is not in our management.

INTRODUCTION

Over the years there have been many presentations related to quality control in SAS programming, more recently risk-based approaches (see [1],[2],[3],[4],[5]). These discussions focus on QC associated with development of programs while ensuring high quality data and reporting for clinical trials. This typically entails two sets of programs: one for production and a second for testing. While the gold standard for testing is independent reproduction, alternative testing approaches are more commonly used and accepted by industry leaders.

Traditional models for DMC support involve the independent SDAC developing SAS code for analysis datasets and Tables, Listings and Figures (TLFs) from raw data. The development and testing of the SAS code is done solely within the SDAC, following the SDAC processes and SOPs.

The focus here is quite different. In this setting, our clients require us to use either their code, or code from another CRO to support a DMC. The "package" received by the SDAC generally contains data, programs, macros (or macro catalogs), blinded output, and sometimes running instructions. Often, the data flow consists of creation of SDTM, ADaM, and TLFs, as DMC reports are viewed by some as simply a subset of clinical study report (CSR) outputs. We refer to these programs as **external code** since it was developed elsewhere. The SDAC is tasked with using the external code developed in a blinded setting to create an unblinded DMC report.

Although this approach is non-traditional, it is becoming more commonly used with larger clients. Even though the SDAC is no longer tasked with developing and testing SAS code, QC measures still exist and should be performed. Because of the increased use of this model, Cytel is implementing processes to ensure the DMC continues to receive a high-quality deliverable necessary to make critical decisions about patient safety and trial integrity.

DEFINITIONS

For clarity, we define the following:

Quality Control: the collection of efforts made to ensure and document that programs produce expected results.

Sponsor: Company conducting clinical trials on a drug or device.

External CRO: Vendor for Sponsor company performing tasks including Data Management (DM) and Statistics.

SDAC (Statistical Data Analysis Center): Vendor independent of the sponsor or external CRO to be unblinded during the study.

Data Monitoring Committee: independent group of experts reviewing unblinded data to assess risk/benefit in order to protect patients and trial integrity, typically includes 2-3 clinicians and 1 statistician.

Open Report: set of TLFs in aggregate, no by treatment arm information.

Closed Report: TLFs by treatment arm.

CONTRACTUAL MODEL

In this setting, the SDAC provides a service that requires interaction of other vendors. Below is a schematic of a typical setting, which includes the sponsor using an external CRO to provide data management and statistical support.

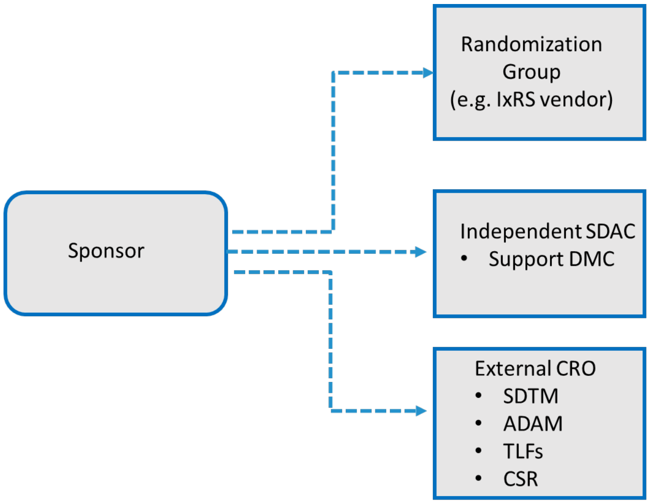

Figure 1: Example Contractual Model

The Sponsor has separate contracts with each vendor, with a desire to use TLF programming from CSR development to support the DMC. Of note, there no contractual link between the vendors themselves. Furthermore, each vendor has data transfer agreements with the sponsor, but not with each other.

Given the external CRO develops code using their internal operating system, processes, and SOPs, quality can be compromised as code gets modified by the SDAC. Additionally, since the code is developed using artificial randomization, quality can also be compromised if programming assumptions deviate from those needed when using the true randomization data.

OPERATIONAL MODEL FOR DMC SUPPORT

When using external code, the operational model, or flow of information, can be quite specific to each project. Below is an example of an operational model for DMC support consistent with the model described in Figure 1.

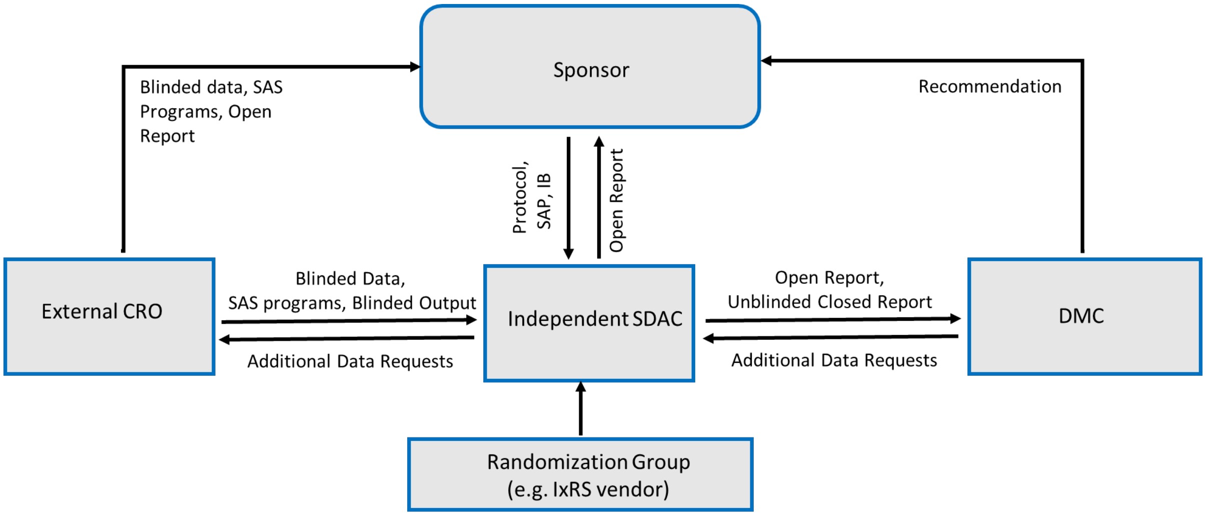
Sponsor
Blinded data, SAS Programs, Open Report
Recommendation
Protocol, SAP, IB
Open Report
External CRO
Blinded Data, SAS programs, Blinded Output
Additional Data Requests
Independent SDAC
Open Report, Unblinded Closed Report
Additional Data Requests
DMC
Randomization Group (e.g. IxRS vendor)

Figure 2: Example Operational Model for DMC Support

We use the operational model to describe the flow of information between the sponsor and different vendors. Focusing on the SDAC, the first step in QC is to ensure that the SAS programs can be run in the SDAC environment. This alone can be quite challenging and is likely a different discussion. The SDAC sometimes receives test code that was used to validate production code. Once the SDAC is able to re-produce the blinded output, unblinding occurs. Last, programs are run on the unblinded data to generate the final Open and Closed Reports which are sent to the DMC. Let's look at these steps in more detail.

A similar model removes the direct interaction between the external CRO and SDAC, whereby the external CRO sends everything to the Sponsor, who then forwards the package to the SDAC. We find this added complexity increases risk to quality as it usually requires the SDAC to make more substantial updates to the code yet with less communication with the developers. The complexity dramatically increases if the programs are developed in a folder structure specific to the external CRO but mapped to a different structure specific to the sponsor.

CRITICAL AREAS FOR QC

Once the SDAC receives the package, the first goal is to reproduce blinded output that was provided by the sponsor. Recall the assumption is that the production code received is fully tested. Our first step in QC is to ensure we obtain the expected results, which are assumed to be the blinded TLFs provided with the package.

Regarding the transfer itself, it is fair to first ask if we should verify a successful transfer. This may be done, for example, with using checksums, but in the absence of a data transfer agreement, it is difficult to establish a reliable process. However, we do feel that risk is mitigated, and we can assume a successful transfer if we are able to reproduce the blinded output.

Installing the Package

Copying the files to the SDAC server may seem like a simple task, but it is critical to maintain the same folder structure that was used for program development. This ensures search paths are maintained as well as any relative file references that may be used in programming.

The most challenging part of installing the package is for the SDAC to determine how to run the external code, typically in batch mode. Most programming environments utilize one or more setup files that may require the use of the AUTOEXEC system option, or are included in each individual program. The setup files nearly always require modifications to accommodate the SDAC environment. The most common modifications are associated with macro variables to define the path on the SDAC network. This is a self-evident correction, and does not alter any programming logic. Other updates may be required to accommodate macro catalogs, ODS style templates, and formats catalogs (which may need to be re-created internally).

Subsequent modifications to individual programs are often but not always required. Much depends on the amount of collaboration between the SDAC and the original developers of the code. It is extremely important to document all changes that were required. This can be reviewed to assess potential impact on quality. Many updates are self-evident, but others are not. This summary of changes can be shared with the code developers to learn and improve for subsequent DMC meetings.

When the files arrive in a folder structure different from program development, getting programs to run successfully is extremely challenging. In our experience, this tends to be a "one at a time" approach to see if programs run successfully.

Quality Control of the Installation

Below are two QC measures that can be considered to ensure programs are producing the expected results on the SDAC server:

1. Electronic comparison of blinded ADaM datasets created by the SDAC compared to those that were provided with the package.
2. Comparison of one or more blinded TLFs, either manually or electronically against the corresponding outputs provided with the package. Electronic comparisons can be done using tools such as MS Word comparisons for RTF files, DOS comparisons of text files, or PDF comparisons with Adobe Pro. We expect some differences, such as the date/time stamps of execution and network paths that often appear in footnotes. Differences in how spaces and dashes may also be found electronically, though we do not consider these relevant.

Every project is different, but we find these options almost always apply. Furthermore, (1) is often skipped with the understanding the if the TLFs match, it implies that the underlying analysis datasets probably match as well.

We recommend the use of electronic comparisons as the resulting files can be saved as documentation that the programs produced the expected results.

It is counter-intuitive to many programmers to suggest modification of SAS code post-validation. This would surely require retesting. However, in this situation, we are essentially "testing" our updates by comparing the results with those provided in the package. Regardless of the number of updates required by the SDAC, if the results match, we conclude that the modified programs continue to generate the expected results.

Quality Control of the Unblinding Process

The unblinding process varies from project to project. It may be as simple as updating the external code of a single program (such as DM) to point to the actual randomization file rather than a dummy file. This is a trivial update, and we are not writing new code. We do not feel additional testing is required given other measures of QC on the unblinded TLFs.

When SDTM data are provided, the SDAC may choose to update the ARM/ARMCD values using the actual randomization file. With this approach, the ARM/ARMCD variables are updated using internal SDAC programming, and recommend following internal SDAC standards for programming and testing of datasets.

Additional QC of the unblinding process may be done after generation of the unblinded ADaM datasets. Often the statistician or a separate programmer will compare the treatment group and stratification variables from the unblinded ADSL directly with the raw randomization file. This also verifies that the sum of the treatment group totals equals the expected total, providing assurance that the randomization file is current and complete.

Quality Control after Unblinding

After unblinding, analysis datasets and TLFs are re-run to create the final Open Report and unblinded Closed Report. Even upon successful execution of the code, additional QC can and should be performed. One or more of the following tasks may be performed to ensure the Closed Report utilizes the actual randomization.

1. Comparison of blinded and unblinded ADaM datasets. Only treatment group (and possibly stratification) variables should change. All analysis datasets should be checked to assure tables subsequently run are done so with the actual randomization.
2. Comparison of Total column to blinded report received with the blinded package. While we expect the treatment group columns to differ, the total column should remain unchanged for most tables.
3. Review the disposition table to make sure subset counts sum to totals.
4. Review tables to ensure column header Ns are consistent with disposition table

Last, we strongly recommend a statistical review should be performed by the independent statistician reporting to the DMC. In this model, programmers are tasked with running code, not developing it. This makes it much more challenging to understand the data. The statistician, however, needs to thoroughly review the outputs for every DMC meeting and likely better understands the data. This statistical review may include the following:

i. Verify exposure summary statistics are consistent with expected dosing per protocol

ii. Verify safety outputs are consistent with a known safety profile

iii. Review of log files for programs with statistical models to look for messages relevant to the modeling

iv. Skeptical review of key safety (and efficacy) tables

v. Visual side-by-side comparison of the current report with the previous report

vi. Assess if observed trends are expected based on previous reports

vii. Investigation of anomalies

Never underestimate the value of a statistical review. Good statisticians often find discrepancies and identify anomalies during and after production programming has been tested. This remains true even if the programming was external code.

DOCUMENTATION

It is simply a good programming practice to document changes to programming code over time. Documenting who did what, and when, ensures production code and output precedes test output, and that the authors have the appropriate training and experience to perform the tasks. Trends point to utilization of source version control but this is challenging to implement. In the model with external code, it may be impossible. Modifications are typically documented in a header of the program itself. Interesting, it is fairly common for the SDAC to receive programs with no program header.

When using external code, our approach is to *document what we do*:

1. All modifications to the external code, even if the updates are trial or self-evident updates. Ideally modifications are listed in a single document. This can be reviewed by SDAC programmers and statisticians to assess potential impact on quality of the outputs. It can also be shared with the original code developers to learn and improve the process for subsequent DMC reports. If, at any point, concerns arise, the SDAC can request that the original authors make updates. Extreme caution should be used if issues arise after unblinding. For this reason, we recommend maintaining separate folders for blinded and unblinded versions of the DMC reports.

2. Retain output for file comparisons, if any. Proc compare output, Word comparisons, or PDF comparisons serve as evidence that such a comparison was performed.
3. Description of the unblinding procedures.
4. Identify and describe QC preformed on unblinded datasets or TLFs. Note that some of this may be documented elsewhere if there are internal SDAC procedures for statistical review of a deliverable.
5. Zip and archive both the blinded and unblinded work areas. If blinded evidences are retained, then retaining the blinded materials generated by the SDAC is less important. However, we believe it is best to retain the folders and blinded outputs in case issues arise after unblinding. Having the blinded programs, logs, and outputs, make discussions with the original developers easier and reduce the risk of accidental unblinding.

As we continue to develop the process of using external code, we continue to evolve with documentation practices. We do this with an open mind, recognizing anything is better than nothing, and nothing will be perfect.

RUNNING TEST PROGRAMS

Sometimes the package with the external code contains both production code as well as the corresponding test code. SDAC programmers and statisticians often ask, ***should we run the test code on unblinded data***?

Before attempting to answer, we recall that an underlying assumption in this operational model is that the SDAC receives external code that is already validated with the current set of data. The only difference in the unblinded data is with treatment group information. While this can introduce risk, it may be acceptable when it is mitigated by other forms of QC as described above.

Individual project teams may consider running external test code if it can be done with reasonable efforts. However, questions remain as to the value gained. We know that not all testing is electronic, and that QC extends beyond comparison of outputs. Is it reasonable for the SDAC formally attest that the test code run on unblinded data did not have any findings? The SDAC is not trained on the SOPs/process or required documentation used where code was developed. We do not want to imply that we validated the output on unblinded data.

We recognize there is inherent risk with this business model. We believe it can be mitigated with other QC procedures, but it can never be eliminated. We recommend being transparent with the sponsor and other vendors.

MINIMIZING CHALLENGES

Whenever possible, a collaboration with the external code developers is essential to minimize required updates and associated risk from them. This can be challenging in the absence of data transfer agreements or contractual obligations. We encourage the SDAC team to meet with the external code developers early in the process to better define expectations, and share experiences with this type of model for DMC support. Discussions in this early meeting may involve:

1. Use of compiled macro catalogs, style templates, etc, especially when code is developed in a different operating system than that of the SDAC
2. Different maintenance releases of SAS
3. Setup files
4. Process to run code in batch
5. Unblinding process
6. Process for creation of a single file deliverable
7. Hardcoding

8. Differences in process for a formal interim analysis

Relationships develop over time. When efforts are collaborative, the SDAC and the external code developers tend to work better throughout the course of the project, and the process becomes smoother with subsequent DMC meetings.

Simplification of DMC report production should also be considered to reduce challenges that arise. While the SDTM-ADaM-TLFs model serves well as a submission model, it is not necessary for DMC support. Incorporating the data flow early can have advantages for the development team, but we find the process can rapidly become complicated for the SDAC when trying to reproduce the process on SDAC servers.

CONCLUSION

While the model of using external code to support a DMC has been debated over the years, current trends suggest an increased use of the model where the SDAC does not develop or test production code. However, as demonstrated in this discussion, quality control efforts are abundant, and quality does not need to be compromised.

We continue to develop the process of using external code to support DMCs, and continue to learn how to improve the quality of the DMC report given limitations of using external code.

REFERENCES

[1] Randall, A. and Coar, W. Assessing Performance of Risk-based Testing, SAS Global Forum 2020

[2] Marrer, Jennier, Risk-based-Program-Validation-for-Clinical-Trial-Analysis-and-Reporting, PhUSE US Connect 2019

[3] Randall, A. and Coar, W. Risk-based Validation in Clinical Trial Reporting: Focus on What Matters Most, PharmaSUG 2018

[4] Combining quality of productivity and efficiency under highly pressure of lacking time – discussion by view of first-time quality, PharmaSUG China 2018

[5] Coar, W. and Randall, A., Data Monitoring Committee Report Programming: Considering a Risk-Based Approach to Quality Control, PharmaSUG 2017

ACKNOWLEDGMENTS

The author would like to thank Benjamin Esterni and David Kerr for their review and insights from the perspective of an SDAC statistician. The author is also grateful Cedric Marchand and Caroline Terrill for their suggestions from a quality and statistical programming perspective.

To Amber Randall and Joshua Sanders, for your review of this paper and willingness to always listen and talk QC, thank you.

CONTACT INFORMATION

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

William Coar
Axio Research, A Cytel Company
William.coar@cytel.com