



Paper SP04

Building a Deeper Bench: Writing ADaM Specifications

Kristen Harrington, Rho, Inc., Durham, NC, USA

Shane Rosanbalm, Rho, Inc., Durham, NC, USA

ABSTRACT

In a world of silos, an organization can get by with having only a few individuals possessing the knowledge to write CDISC-compliant ADaM specifications. While silos do allow folks to become really good at one thing, it doesn't necessarily lead to a better quality study overall. A better model is to have the same programmers responsible for not just one aspect of a study (SDTM or ADaM or TLFs) but rather have them be responsible all aspects of a study (SDTM and ADaM and TLFs). This paper will focus on Rho's efforts over time to expand ADaM spec writing expertise from the core group of ADaM early-adopters to the broader programming group as a whole.

INTRODUCTION

This paper is about the task of writing ADaM specifications. More specifically, it's about how to train folks to be able to write ADaM specifications. In the early days of ADaM standards, sponsors were slow to embrace them. This meant the volume of this type of work was low enough that you could just about get away with handing someone a copy of the Implementation Guide and turning them loose, accepting the corresponding inefficiency. Now, ADaM is the strong default and new employees need to be quickly and efficiently trained to be fluent in ADaM. In this paper, we will provide an overview of the process we use to train folks how to write ADaM specifications.

CONTEXT

ADaM specifications are not written in a vacuum. If you were to make a list of steps that surround ADaM specifications, it might look something like this:

- Create protocol
- Create CRFs
- Create SAP/TLF mocks
- Create SDTM specifications
- Program SDTM datasets
- **Create ADaM specifications**
- Program ADaM datasets
- Program displays
- Create define packages
- Fight with P21E

As you can see, there are a lot of elements that lead into the ADaM specifications, as well as a couple of key downstream outputs that are greatly impacted by the specifications.

TRAINING OVERVIEW

The training program that we developed involved writing specifications for an actual study (no test data here). As such, there were a few pre-requisites that each trainee needed to meet.

- Programmed ADaM dataset for a prior study
- Wrote SDTM specifications for the current study
- Perused the ADaM model and IG documents

We then assembled trainees for in-person trainings on the following topics.



- Big picture overview of ADaM model and IG
- How to informally annotate SAP and TLF mocks
- Leveraging Rho's in-house gold standard metadata
- Common conformance issues
- Define and ADRG basics

Interspersed during and subsequent to training, the trainees engaged in the following activities.

- Write specs
- Get mentor feedback
- Program specs
- Run P21E reports
- Frequent formal check-ins
- On-demand group chat

RESOURCING

In the early days of data standards, Rho had programmers responsible for SDTM specifications and statisticians responsible for ADaM specifications. As the volume of data standards work has increased over time, we have transitioned to having programmers responsible for both SDTM and ADaM; programmers are just more efficient. Additionally, the more familiarity one has with the SDTM specifications for a particular study, the easier it will be to write the ADaM specifications for that study. While it is possible to produce quality output using two separate individuals (one for SDTM specs, another for ADaM specs), it is usually more efficient to have one individual be responsible for writing both the SDTM and ADaM specifications for a study. Sometimes logistics will conspire against you (eg, a rescue study). But, if you have the luxury of time, you should assign one person to handle both SDTM and ADaM specifications.

GOLD STANDARD

Building off of the Implementation Guide and years of ADaM specification creation experience, the Gold Standard for ADaM specification provided a familiar starting point for new trainees and experienced spec writers. The Gold Standard consisted of common variables, definition considerations, and information required for the define file. However, ADaM metadata is more than just a spreadsheet containing variable names, labels, and definitions. The information must be stored and structured such that it can be easily extracted when applying attributes to the SAS datasets as well as when creating the define file. For this reason, we developed a metadata tool which worked synchronously with other in-house tools or as a standalone tool.

TRAINING

Once our infrastructure was in place, we began developing training geared towards programmers. Ideally, a single programmer generates both SDTM and ADaM specifications for a study. Our training plan development began with this premise, and then expanded to other resourcing scenarios. Further, based on other successful trainings, we determined that group face-to-face presentation trainings, accompanied by hands-on tasks, working with a ratio of 1:2 (one trainer to two trainees) was the most efficient and effective training approach. Under the assumption that the trainees were responsible for the SDTM specifications, familiar with the protocol for each study, and had at least a rudimentary familiarity with ADaM having programmed it previously, training began with in-depth discussions of the Statistical Analysis Plans (SAP) and Mocks TLFs. The focus of these discussions was interpretation and feedback. We began with analysis populations, documenting the definitions and ADaM variables within a copy of the SAP document. This reinforced the link between the SAP and the ADaM specifications, and annotating a copy of the SAP (either manually or electronically) ensured nothing was missed.

Example of electronic annotation:

4.1 The Safety Population

The Safety population will be used for safety analyses. The Safety population comprises all subjects who received at least one dose of study treatment.





We continued through a wide variety of SAP topics, outlining the types of variables and data structures that were commonly utilized in each. Trainees made notes on the variables that they would need to create, and if necessary, sometimes an outline of a definition or data source to be used in the derivation.

Consistency was encouraged throughout the training and the Gold Standard spreadsheet. A good example would be analysis flags. We wanted to reinforce the idea that when practical the ANLxxFL variables should mean the same thing in all ADaM datasets. For example, it would be expeditious to use ANL01FL and ANL02FL for prior and concomitant medications. However, this would not support consistency across ADaM domains. Instead, we promoted the use of new flag variables, PRIMEDFL and CONMEDFL, in the ADCM Gold Standard.

Once the SAP had been reviewed at least once, the Mock TLFs were reviewed. This critical review focused on ensuring ADaM, or the annotated SAP at this stage, fully supported the Mock TLFs. Trainees reviewed the document to ensure any necessary or helpful flags were included. This included potential denominator flags for subgroups or other subsets within the analyses. Further, the Mock review concentrated on ensuring the appropriate structure was present in the ADaM specifications. For example, a table summarizing Hy's Law may best be supported via an independent analysis dataset specific to Hy's law parameters only. As another example, the difference between ANLxxFL and CRITxFL was thoroughly described.

Once a copy of the SAP had been thoroughly annotated based on the SAP and Mock TLF document reviews, the Gold Standard spreadsheet was introduced. Under our initial assumption that all of the trainees were familiar with ADaM from a programming perspective, the Gold Standard spreadsheet was introduced as a specification tool to provide consistency across domains, highlight strict Implementation Guide rules, and make suggestions for rules that are more interpretable. Further, as the Gold Standard spreadsheet was created to integrate with our in-house define file creation tools, the spreadsheet became an integral part of the full ADaM submission process. This full ADaM submission scope led us to expand the training beyond ADaM specifications and into the define file and Pinnacle 21 battle simulations as well.

CONCLUSION

In training programmers to create ADaM specifications, we have found efficiencies in working with programmers who have experience with ADaM programming from past studies and authorship of the SDTM specifications for the current study. With trainings focused on informally annotating the SAP and TLF mocks, leveraging Rho's in-house gold standard metadata, common conformance issues and the define and ADRG basics, we found several different training formats were appropriate. Pairing the programmer trainee with an experience Biostatistician in a mentor relationship provided a supportive and efficient training relationship. We also found that a training cohort, training several programmers simultaneously, organically created a supportive group for the trainees to use as further resources for questions and brainstorming.

CONTACT INFORMATION

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

Kristen Harrington
Rho, Inc
Kristen_Harrington@rhoworld.com

Shane Rosanbalm
Rho, Inc
Shane_Rosanbalm@rhoworld.com

Brand and product names are trademarks of their respective companies.