

Worst CDISC Implementation Processes – How to Avoid Some Very Basic Mistakes in SDTM, ADaM, Define.xml and Reviewer’s Guide Production

Hannes Engberg Raeder, PRA Health Sciences, Mannheim, Germany
Michael Reich, PRA Health Sciences, Mannheim, Germany

ABSTRACT

These days, most if not all data professionals in this industry, whether they work for a pharmaceutical company or a CRO, have a solid awareness of CDISC and regulatory agency data submission requirements. However, not all organizations/study teams are aware of the pitfalls that can result from inadequate processes surrounding the implementation of these requirements – SDTM, ADaM, define.xml and reviewer’s guides. This paper describes a number of real cases related to the processes of implementing the standards referred to in the standards catalogues published by the FDA and PMDA. Examples described come from both CRO study teams, sponsor’s internal programming teams and the relationship between CRO and sponsors. The focus will be on how to identify and avoid these dangers that can result in costly delays for the sponsor and significant overburn for the CRO.

INTRODUCTION

In this paper we do not focus on sleek technical tools and implementations, but rather on simple solutions on how to avoid devastating process blunders any CRO or sponsor can encounter while building their data package.¹ Based on examples collected through the years at CROs and sponsors, we want to look at the potential hazards related to four aspects surrounding the implementation of CDISC standards: The production of define.xml and reviewer’s guide, more precisely how and when we do produce these items. Secondly, we also discuss the advantages of looking for and addressing issues encountered early during the generation of data package outputs. We also look at some very useful thinking linked to the overall review of a package including not only CDISC-compliant data, metadata and reviewer’s guides but also the tables, figures and listings. Finally, we look at the differences between FDA and PMDA compliance and CDISC compliance. As we will see, building CDISC-compliant data packages is not necessarily the same thing as following the standards currently supported or required by FDA or PMDA. With that said, this paper does not claim to provide ultimate solutions, but we do believe that taking these four aspects into consideration will help any programming team significantly reduce quality, timeline, and budget risk when creating their submission-ready data packages

WHY YOU SHOULD NOT BE CREATING DEFINE AND REVIEWER’S GUIDE AT THE END

Producing the content of the define.xml and the reviewer’s guides at the end is not only a risky and inefficient strategy that will likely result in delay, mistakes and over burn on the project, but is contrary to FDA’s opinion about define.xml which “...is considered arguably the most important part of the electronic dataset submission for regulatory review.”² The concrete danger of submitting an incorrect define.xml file and/or reviewer’s guide is in a best case scenario a delay, as the sponsor needs to spend time on resolving queries from the regulatory reviewers, while on the other side of the spectrum we have the looming threat of a rejected submission³. In the two examples below we look at one scenario involving define.xml and another involving the reviewer’s guide, as well as a proposed solution.

MISSING OR INCORRECT COMMENTS IN DEFINE.XML

For a large phase III study it was decided to push back production of the define.xml to a few days before sending the final data package to the client. At the beginning of the project an empty shell for the define.xml was created with the comments field for the complex derivations left empty, as can be seen in Figure 1. The plan was to populate these fields when most of the programming work had been completed. However, when the time came for the team to start working on the define.xml, most of the original programmers had left the project in order to work on other tasks, and to

¹ With data package we mean the typical components of deliverables compliant to either FDA or PMDA’s standards catalogues, which means SDTM, SEND, or ADaM accompanied with corresponding define.xml 2.0 and Study Data Reviewer’s Guide or Analysis Data Reviewer’s Guide (SDRG and ADRG).

² FDA’s Study Data Technical Conformance Guide Version 3.3, p. 18.

³ PMDA’s Study Data Validation Rules (2015-11-18) <https://www.pmda.go.jp/english/review-services/reviews/advanced-efforts/0002.html>

PhUSE 2017

be able to populate these fields and generate the define.xml content the remaining programmers had to re-visit raw data, SDTM data, and the underlying SAS programs.

| Variable | Label | Type | Length / Display Format | Controlled Terms or Format | Source/Derivation/Comment |
|----------|--------------------------------------|---------|-------------------------|----------------------------|---------------------------|
| AWRANGE | Analysis Window Valid Relative Range | text | 9 | | |
| AWTARGET | Analysis Window Target | integer | 8 | | |
| AWTDIFF | Analysis Window Diff from Target | integer | 8 | | |
| AWLO | Analysis Window Beginning Timepoint | integer | 8 | | |
| AWHI | Analysis Window Ending Timepoint | integer | 8 | | |

Figure 1.

The consequences were a delay for the client and additional costs to bear for the CRO, as the CRO team had to spend time on regaining an understanding of the data that they already had previously had at one point in time, but which they had failed to document properly at that time.

In the second example, Figure 2, the comments for the define.xml had been copied over from the programming specs, cheap and quick. However, as the lead programmer started to review the define.xml just days before sending it to the client, several issues were discovered: The comment describing the derivation of EXDOSE in the Exposure domain did not reveal an algorithm that could be understood without also understanding the raw data and possibly also the code used (for example, it was not clear what *table* or *Table* are to either the programmer or anyone else reviewing the define.xml).

| Variable | Label | Key | Type | Length | Derivation/Comment |
|----------|-------------------------|-----|---------|--------|---|
| EXDOSE | Dose per Administration | 6 | integer | 2 | IF tabl >10 set to 4, else if Table not missing, set to 6. else set to table X 12 |
| EXDOSU | Dose Units | | text | 2 | Derived from ARM, ARMCD - equal to mg |
| EXDOSFRM | Dose Form | | text | 7 | |

Figure 2.

When incorrect or completely missing comments and other metadata is piling up during the final review of the define.xml, it will jeopardize not only the quality but also the timelines, as programs and data often need to be revisited during this process.

THE INCOMPLETE REVIEWER'S GUIDE

If the define.xml is the blueprint of the database (SDTM, SEND or ADaM), the reviewer's guide is the user manual. It is indispensable for anyone who wants to understand the tabulated and analysis data, whether it is the internal senior statistical reviewer at the CRO, the lead programmer at the sponsor, or the reviewer at the regulatory authority. Often its content can be produced after completion of the study, especially sections relying on protocol input or the annotated CRF design. Just as with the define.xml, though, some of the information that needs to go into the reviewer's guide will be just as time consuming to re-construct later.

In the next example we look into section 5 of the ADRG. In short; this section provides a mandatory inventory of the analysis datasets employed in the study. In particular section 5.2 enables the author to not only describe all the datasets (which is required), but also to describe analysis datasets requiring more detailed explanation⁴. In Figure 3 below, the project statistician wanted to gain a better understanding about the time to event analysis data in ADTTE. However, the description in the reviewer's guide only stated the obvious about the content and nothing about the particular data issues related to the underlying data used in generating the dataset. This was communicated to the programming team as a finding. In order to describe this, the programmer working on the study had to investigate the raw data, the SDTM data, and the analysis program used. This led to a substantial delay in finalization of the data package.

⁴ PhUSE's Analysis Data Reviewer's Guide Completion Guidelines Version 1.1, p. 18.

5.2.2 ADTTE – Time to Event Analysis Dataset

The time to event dataset was used to support the primary study endpoints.

5.2.3 ADLBCHEM – Chemistry Results Analysis Dataset,

These laboratory datasets were used primarily to support shift table analysis based on toxicity grade.

Figure 3

For the same study some concerns were also raised concerning the SDRG, in particular the Data Conformance Summary. This section is required and will hold any compliance issues found using validation tools like Pinnacle 21 Community Version or other software⁵, along with explanations why these issues will be a part of the final package. During the review the lead programmer discovered that only the Diagnostic Message, Severity and Count columns had been populated (copied from Pinnacle 21 report) and the Explanation column had been left empty (to be populated later), see figure 4 below. Although many of the issues reported in the Pinnacle 21 report can be easily found in the data and, if they cannot be resolved due to database lock or being false positives, they can usually be described with few lines of text. However, as with the define.xml examples above, it is the volume of unresolved/not explained issues that will cause the delays to the project.

4.2 Issues Summary

| Dataset | Diagnostic Message | Severity | Count | Explanation |
|---------|---|----------|-------|-------------|
| AE | EPOCH value not found in 'Epoch' extensible codelist | Warning | 28 | |
| AE | Missing End Time-Point value | Warning | 3 | |
| AE | Model permissible variable added into standard domain | Warning | 1 | |
| AE | Permissible variable with missing value for all records | Warning | 6 | |
| AE | No Treatment Emergent info for Adverse Event | Warning | 28 | |
| CM | CMDOSFRQ value not found in 'Frequency' extensible codelist | Warning | 1 | |

Figure 4

SOLUTION

The define.xml (applies to SDTM, SEND and ADaM) is an integral part of the data package, without which it would just be a set of tabulation or analysis data telling the reviewer only half of the truth. In theory the define.xml content should be driving how we map the data, design the algorithms used in ADaM, or which codelist values we want to display in the data, to name a few examples. In reality the relationship might be a bit more diffuse, but treating the data and the metadata as two independent entities is a recipe for failure. What we suggest here is to set-up the define.xml production at the same moment the data specification and programming starts. This can be done in many different ways and is of course dependent on the respective organizations technical infrastructure.

The cheapest and simplest solution is the good old spreadsheet set up in way that it can hold any programming specific information (not to be included in final define.xml) as well as the required information needed for the define.xml. There is also a great deal of overlapping between programming specifications and define.xml, which means that a great deal of the specifications can be used in the define.xml. The key to a resourcefulness implementation is to set up the spreadsheet in a way that the information can be machine readable to generate the XML database as well as hold the programming specifications.

Figure 5 below illustrates a simple spreadsheet solution which enables the specification author and the programmer to have access to the programming specific information and the actual define.xml content at a glance. The second column from the right is typically the text that would end up in the final XML database. We want that text to be accessible to anyone involved in the study (statistician, medical writers and external reviewer to mention a few); that means in plain English without too much technical jargon (SAS code) or raw data references. The rightmost column (called *Additional comments*) will hold specific programming instructions that will not be included in the final define.xml. Most of the time the comments going into the define.xml will be sufficient as programming instructions; however, there are situations when it is advantageous to add more technical details. As can be seen for the variable EXSPID it was deemed that

⁵ PhUSE’s Study Data Reviewer’s Guide Completion Guidelines Version 1.2, p. 10.

PhUSE 2017

the define comment would not be sufficient for the programmers implementing the mapping, hence the additional comment was added. In the row below, for EXTPT is more straightforward, as it comes from the CRF data with an underlying codelist, so no further comment in the define was necessary. Nevertheless, the programmer was instructed to use a certain macro to populate that variable in the additional comments. The major benefit is that the define.xml content can be developed and also reviewed simultaneous with the programming specifications.

| Variable | Label | Type | Length | Codelist / Controlled terms | Source | Origin | Mandatory (Yes/No) | Role | Comments | Additional comments |
|----------|----------------------------|------|--------|-----------------------------|---------------------|----------------------|--------------------|------------|---|--|
| STUDYID | Study Identifier | text | 10 | | | Protocol | Yes | Identifier | | |
| DOMAIN | Domain Abbreviation | text | 2 | DOMAIN | | Assigned | Yes | Identifier | | |
| USUBJID | Unique Subject Identifier | text | 30 | | raw.dose.pt | Assigned | Yes | Identifier | Study identifier, site identifier, and subject identifier, concatenated and separated by hyphens. | Site Identifier is :XX' Subject Identifier: concatenate '000' with subjid (for example: '000101') |
| EXSPID | Sponsor-Defined Identifier | text | 1 | | | Derived | No | Identifier | Sequential number identifying records within each USUBJID for supplemental dosing data | sort raw.suelup (whenever the variable up01 ~= blank) by PT and UDDTC and assign number per subject by UDDTC |
| EXTPT | Planned Time Point Name | text | 15 | TPT | raw.dose.time point | CRF Pages 17, 22, 45 | No | Timing | | Use the compiled macro map_vis_tp, using the visit tab as input |

Figure 5

Reviewer's guide content is driven by the study tabulation data or the analysis data and cannot be finalized as long as there are outstanding updates to core variables, dataset descriptions, or data issues. However, the usefulness and accuracy of the reviewer's guide depends heavily on the author's in-depth understanding and knowledge of the study and the study data. Again our go-to solution is the spreadsheet: Make it a habit to set it up at the beginning of the project to collect all data-related information and issues that you might need to explain in either the SDRG or the ADRG.

WHEN AND HOW YOU SHOULD LOOK FOR ISSUES IN THE DATASETS

Even if the advice presented in the previous chapter is followed and define packages are created and maintained in parallel with the tabulation and analysis datasets themselves, this alone will not protect the CRO or sponsor from another common mistake encountered far too often during the creation of the data package. This section takes a look at the problem of postponing various reviews and checks that could be done earlier, or even deciding to only properly fix identified issues later down the road. Both situations regularly result in unexpected extra-work requirements, which in turn can lead to delays or even database unlocks. To make matters worse, not properly addressing late-identified issues related to CDISC compliance can even lead to a rejected submission⁶.

UNDERUTILIZING AVAILABLE VALIDATION TOOLS

These days Pinnacle 21 is a widely known and highly appreciated validation tool. This open source software provides invaluable support to study teams in creating CDISC-compliant SDTM and ADaM data and define.xml files, or in identifying various kinds of potential data issues. The use of Pinnacle 21 becomes all the more essential as both the FDA and PMDA increasingly rely on Pinnacle tools to conduct their own validation on data provided by sponsors. Considering this, it is all the more surprising that the tool is very often still not used incorrectly or as it should be, as is shown in the following examples.

In the first case, the programming team responsible for a phase III study reached the point of producing final tables, figures, and listings, using an underlying ADaM database that was double-programmed and lead-reviewed. As the project reached its end and the associated documentation was to be finalized as well, Pinnacle 21 was run on the ADaM database for the first time. What this late validation step revealed (partially shown in Figure 6) was the necessity to not only implement extensive updates to the ADaM database, but also to update close to all table, figure, and listing programs to accommodate the changed metadata-structure of the input ADaM datasets.

The resolution of these issues postponed the finalization of the study by two months, severely decreasing the available time to properly include this study into the corresponding FDA NDA integrated analysis and report.

⁶ PMDA Validation Rules 2015-11-18

PhUSE 2017

| Issue Summary | | | | |
|---------------|------------------------|-------------------------------|---|----------------|
| Source | Pinnacle 21 ID | Publisher ID | Message | Severity Found |
| ADAE | | | | |
| | AD0042 | 42 | *TM does not have the ADaM required SAS time format | Error 2 |
| | AD0269 | 269,270,271 | TRTEMFL flag value is not Y or null | Error 111 |
| | AD0279 | 279,282 | AESEVN value != 1, 2, 3, or null | Error 235 |
| ADEG | | | | |
| | AD0018 | 18 | Variable label mismatch between dataset and ADaM standard | Error 5 |
| | AD0047 | 47,49,50,51,5 | Required variable is not present | Error 1 |
| | AD0110 | 110 | Inconsistent value for AVISIT | Error 149 |
| | AD0256 | 256 | USUBJID value does not exist in the ADaM ADSL domain | Error 21561 |
| ADLB | | | | |
| | AD0018 | 18 | Variable label mismatch between dataset and ADaM standard | Error 4 |
| | AD0047 | 47,49,50,51,5 | Required variable is not present | Error 1 |
| | AD0110 | 110 | Inconsistent value for AVISIT | Error 196 |
| | AD0141 | 141 | Inconsistent value for PARAM within a unique PARAMCD | Error 1531 |
| | AD0146 | 146 | Inconsistent value for PARAM | Error 1531 |
| | AD0185 | 185 | Illegal variable name: y is not in [1-9] for PARCATy(N) | Error 1 |
| | AD0192 | 192 | Illegal variable name: y is not in [1-9] for SHIFTy(N) | Error 1 |
| | AD0196 | 196,197 | Required PARAM value is null | Error 260 |
| | AD0196 | 196,197 | Required PARAMCD value is null | Error 260 |

Figure 6.

In the second example, the lead programmer did run Pinnacle 21 on the SDTM database rather early, but dismissed the majority of identified issues, incorrectly attributing them to the ongoing status of the study and assuming that the issues resulted from incomplete and dirty data and that the issues would resolve themselves by database lock. Only after database lock occurred and the first TFL deliverable was approaching did the programmer really begin to dive deeper into the meaning of the validation report findings to complete the SDRG documentation. The programmer was then not only confronted with a number of SDTM implementation errors that had gone undetected, but also with data issues that could have been easily resolved via queries by data management prior to the lock.

The draft tables, figures, and listings shipment had to be postponed by a couple of weeks to ensure that the SDTM database was truly compliant and that the necessary updates had no ill effect on the ADaM database and table, figure, and listing outputs that had already been programmed on the basis of the non-compliant SDTM data. Further, extra-time was needed to accommodate the data issues in the SDTM and ADaM programming and ensure proper documentation in the corresponding reviewer guides.

CUTTING SHORT ON LEAD REVIEWS

While Pinnacle 21 helps to improve CDISC compliance and data quality if used correctly, additional manual review is still necessary in order to ensure adequate data compliance and data quality. Usually, this is done in the form of a lead programmer review shortly before a shipment of outputs created by the programming team. However, limiting this review to only this timeframe is another increases the risk of delays and extra-work that could have been prevented had certain manual review steps been performed earlier in the process. In addition, missing or vague guidelines on what a lead review is supposed to cover or how it should be done can also lead inconsistent results.

For another phase III study, the lead programmer seemingly did everything right. The lead review showed that the output logs were clean, both SDTM and ADaM passed their respective Pinnacle 21 runs, and even the tables, figures, and listings looked fine. But all the lead essentially did was rerun the same granular QC steps that had already been implemented and conducted by the programming team; holistic review steps such as cross-checks between tables and the underlying data or manual reviews of sample records of each dataset were not performed. The sponsor counterpart unfortunately also did not do these kinds of deeper reviews and the study team eventually proceeded to move towards finalization of the table, figures, and listings.

At the same time, a second CRO contracted with the creation of integrated summaries of safety and efficacy for the FDA noticed various discrepancies while comparing their integrated analysis results with the study level tables. Two of these are described as examples below to illustrate how they were found and thus could have been prevented.

The first one was related to a table summarizing events derived from available finding data (vital signs, ECG, lab results, etc.). The cross-check of the underlying analysis dataset against the table revealed that the study-level table included both pre- and post-dose events – the latter up to 30 days after last dose – even though the SAP specified that only events occurring after dosing and up until 7 days after last dose were to be included in the analysis. This went unnoticed as all parties reviewing the output implicitly assumed that the data selection for this table had been done as specified and previously verified to be correct by the lead programmer: the review focused solely on the table outputs and ignored the link between the underlying ADaM datasets and table outputs.

PhUSE 2017

The second example was related to the identification of treatment emergent events. Both SAP and define.xml indicated that no imputations were to be done and defined treatment emergence the same way, including that an event with a missing start time but a start date on the day of treatment should be considered treatment emergent. It went unnoticed that both main and QC programming mistakenly imputed the time portion of ADAE ASTDTM (analysis start date/time) to 00:00:00 if a start time was missing in AE.AESTDTC (SDTM AE start date/time), leading to all these events being considered pre-treatment events. As can be seen in figure 7, this could have been quickly noticed as ASTDTM is always populated, even when ASTTM (analysis start time) is empty considering that there should be no imputations.

| TRTSDTM | AESTDTC | ASTDT | ASTTM | ASTDTM | TRTEMFL | PRETRTFL |
|---------------|------------------|-----------|-------|------------------|---------|----------|
| 02NOV15:11:28 | 2015-11-02T11:50 | 02NOV2015 | 11:50 | 02NOV15:11:50:00 | Y | |
| 22OCT15:14:30 | 2015-10-23 | 23OCT2015 | | 23OCT15:00:00:00 | Y | |
| 04NOV15:15:13 | 2015-11-04T15:00 | 04NOV2015 | 15:00 | 04NOV15:15:00:00 | Y | |
| 13JAN16:13:23 | 2016-01-13 | 13JAN2016 | | 13JAN16:00:00:00 | | Y |
| 08AUG15:15:01 | 2015-07-27T15:00 | 27JUL2015 | 15:00 | 27JUL15:15:00:00 | | Y |
| 20OCT15:13:07 | 2015-10-22 | 22OCT2015 | | 22OCT15:00:00:00 | Y | |
| 07JAN16:15:28 | 2015-12-25T10:00 | 25DEC2015 | 10:00 | 25DEC15:10:00:00 | | Y |
| 08JUL15:15:26 | 2015-07-09T08:00 | 09JUL2015 | 8:00 | 09JUL15:08:00:00 | Y | |
| 01JUL15:10:10 | 2015-07-02 | 02JUL2015 | | 02JUL15:00:00:00 | Y | |
| 04MAR16:08:53 | 2016-03-04 | 04MAR2016 | | 04MAR16:00:00:00 | | Y |
| 01JUN15:13:00 | 2015-06-01T10:58 | 01JUN2015 | 10:58 | 01JUN15:10:58:00 | | Y |
| 17NOV15:15:38 | 2015-11-17T16:24 | 17NOV2015 | 16:24 | 17NOV15:16:24:00 | Y | |

Figure 7

These and other issues required updates to different derived dataset, table, figure, and listing programs and more thorough re-reviews to ensure no further quality issues went unnoticed. This ultimately led to a delay in the finalization of this study by a month and posed a risk to the timeline for the submission to the FDA.

SOLUTION

To prevent issues like the ones described above, it is important for the lead programmer to internalize a small set of rules: Review early and often, exhaustively investigate what is found, immediately take action to correct issues, and properly document issues that will not be fixed (be it for an upcoming transfer or for the remainder of the project).

Each deliverable (like the complete SDTM database with define.xml) can be broken down into smaller components that are to be reviewed the moment they are ready; for example, a group of selected SDTM domains like EX, DS, DM, and SE. The review should not only encompass a verification that the applied QC processes were done correctly (i.e. logs are clean, output passes validation tools, etc.) but also aim to include a manual review on a small sample of content to ensure that the presented results are as expected. This ensures that issues are found very early in the process with ample time to appropriately address them, at a point in time when the team that created the corresponding outputs still has fresh memories of what they just worked on. It is also advisable to review ADaM dataset specifications the moment they are ready for programming, be it only to ensure that critical information available to the lead programmer that might not be so obvious in the SAP was not missed by the specifier.

Whenever something is found during the reviews, especially when it comes from validation reports or concerns potential data issues, it is important to thoroughly investigate these. The ultimate goal of this investigation is to enable the issue reviewer to accurately describe the nature and reason of this issue. This ensures that the issue can be sufficiently communicated to the party responsible for fixing it (for example data management), but also facilitates proper documentation in case it cannot be fixed and will remain in the final database.

While investigating and subsequently fixing the identified issues require a certain time investment by the lead programmer and the team, it is essential to not postpone this. As illustrated in the earlier examples, waiting with these items until the very end can have serious timeline and/or budget consequences, making it crucial to immediately query what must be queried and fix what can be fixed.

Lastly, it is important to continue proper maintenance of the associated define documentation at all times. This does not only apply to documenting changes in the SDTM and ADaM databases in order to address identified issues, but also to documenting issues that are not going to be fixed (be it just for the upcoming transfer, or ever).

KNOWING WHEN TO SIGN OFF WHAT

Historically it might have been justified to complete the SAP, which the programmers then would translate into safety and efficacy outputs (tables, figures and listings), and then the statistician at the CRO or study team at the sponsor would indirectly approve the underlying data based on the tables, figures and listings they reviewed. The underlying data would not receive any kind of formal review. At best the data would be in a sponsor defined data structure that would actually enable anyone to search for specific data points, but typically it would be a mixture of derivations in datasets and duplication of algorithms in the table programs and even the developers of the code for the raw databases

PhUSE 2017

and outputs would struggle to follow the data flow after some time. As no regulatory reviewer wanted to see the data, this used to be the state of things for better and worse.

In today's global landscape, most CROs and sponsor have adopted CDISC standards; which makes sense as new drug applications are not only submitted in countries with no data review requirements, but are also part of global programs, i.e. the data from one or more studies are likely to be submitted to more than one regulatory agency, e.g. to both the FDA and the PMDA as well. This means that the way we traditionally viewed the production process from a completed SAP to completed tables, figures and listings is inadequate. Yet, there are CROs and sponsors lacking awareness in this area and continue to ignore the necessity of compliance reviews. This strategy, we would say, is for the fearless ones and might result in situations like the following:

The outputs (the actual tables, figures and listings) are reviewed and accepted, however during the final review of the SDTM and ADM data, the define.xml data, and the reviewer's guides, several compliance issues are found. It might still be that the results as presented in the outputs are correct, but the compliance issues are so severe that the data package cannot be submitted to either PMDA or FDA without updates made to the data. So, in order not to put the submission at risk, the only reasonable option is to make the required updates, and if the updates also include changes to SDTM datasets and ADaM datasets there is a risk that the numbers in the outputs change. It is possible to compare old sets of SDTM and ADaM datasets and outputs programmatically to ensure no unintended discrepancies arise, but this strategy can still be time-consuming and result in an inconsistent dataflow or compromised traceability, as the actual outputs were created prior to the dataset. Either way, reviewing the outputs again or comparing them to identify the differences, we are looking at unnecessary work.

This is exactly what happened to a mid-size sponsor who was preparing a PMDA-submission. The study team received the tables, figures and listing, which were thoroughly reviewed, and the sponsor statistician and medical experts signed off on the outputs and gave the statistical programmers the go to prepare the data package for submission. As the SDRG and ADRG were reviewed, some hints that there might be problems with the data started to surface. The programmers ran Pinnacle 21 together with other in-house tools and discovered a long list of CDISC compliance issues, and neither the SDRG or ADRG had been properly set up to reflect this. The programming team was faced with two (in our opinion bad) options; either re-do the complete review of the outputs after the data had been updated to address the issues, or compare all the outputs and send only those which had changed for a second review. In this case, the submission luckily was not delayed, but the programming team had to spend many additional hours to account for this.

SOLUTION

The simple solution here is to adopt the validation process to replicate the actual creation of the building blocks in a complete package consisting of SDTM, ADaM and metadata. In an ideal world, illustrated in figure 8 below, we would want to catch as many data issues as possible already during the data entry phase (see step 1) – which would make things easier as we move on to the next big hurdle; the SDTM mapping. In step 2 we validate the SDTM package, that is the datasets and the define.xml and if possible at least a draft of the SDRG. Even if SDTM mapping and ADaM programming usually got some degree of overlapping, we would recommend validating data and define.xml content as soon as possible for each completed SDTM domain. Finally, in step 3, we validate the analysis datasets with associated metadata; define.xml and ADRG. At this stage we should also have completed the full validation of the SDTM package.

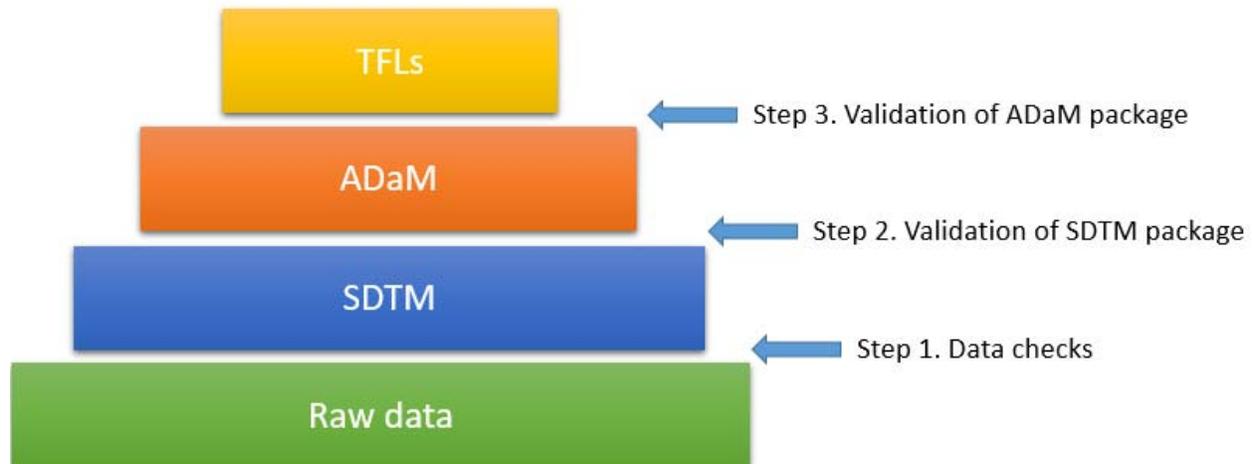


Figure 8.

Following this principle and completing one block after the other will not guarantee that the TFLs will be error-free, but it will prevent unnecessary re-work based on last-minute CDISC compliance issues. What is also worth pointing out here is that this validation model will be more efficient if the process enhancements discussed above are followed.

PhUSE 2017

Finally, there is strong evidence in the guidelines⁷ from these two regulatory agencies, FDA and PMDA, that CDISC standard-compliance is validated before the actual safety and efficacy results are evaluated. With that said there is no reason why the study team should not adopt the same approach.

KNOWING THE PURPOSE OF YOUR CDISC EFFORTS

Indisputably the CDISC standards, and also PhUSE's standardization work, have come a long way and gained recognition at the FDA, PMDA and EMA⁸. Not only that, but the majority of all sponsors (with in-house statistics and programming capacity) and CROs have standardized their data collection, mapping and analysis work to be in line with the models offered by CDISC. However, CDISC compliance is not automatically the same as CDISC compliance as expected by the FDA and PMDA. Two agencies who have published standards they support and require to be followed accompanied by validation rules⁹. Due to time it takes to implement news checks and train staff the standards catalogs lag behind the most recent published guidelines by CDISC. Not a big deal one would think.

Recently, in the turmoil of preparing a data package for the PMDA, the ADRG author realized that several of the severe issues highlighted in the Pinnacle 21 report were in fact related to changes that had occurred between ADaM Implementation guide 1.0 and 1.1. The analysis datasets had been designed using ADaM implementation guide 1.1, ADaM Structure for Occurrence Data 1.0, and the most recent publication of CDISC Controlled Terminology. Although the implementation of the abovementioned implementation guides for ADSL, BDS datasets and event dataset was flawless, it still presented the sponsor with some risks to continue with the current data package. It had already been communicated to the recipient (the regulatory agency review team) of the data package that it would be compliant with the data standards catalog and PMDA's Technical Conformance Guide on Electrical Study Data Submission which requires ADaM Implementation Guide 1.0 and not 1.1. Below, in the figures 9.a-c, are some of the issues as reported by Pinnacle 21, they had to correct.

| | | | | | |
|--------|--|----------------------------------|--|----------|-------|
| AD0047 | 47,49,50,51,52,55,71,72,88,89,90,194,195,257,260,261,262,278 | Required variable is not present | Variables described in ADaM as Required must be included in the dataset. | Presence | Error |
|--------|--|----------------------------------|--|----------|-------|

Figure 9.a

The error in figure 9.a occurs because between ADaM Implementation Guide 1.0 and 1.1, TRTP was dropped as a required variable in the Basic Data Structure (BDS), consequently a BDS dataset is compliant in 1.1 without TRTP. Unfortunately for the sponsor, in 1.0 this variable is required, hence Pinnacle 21 flagged this as an error.

| | | | | | |
|--------|----|---|--|----------|-------|
| AD0018 | 18 | Variable label mismatch between dataset and ADaM standard | Variable Label in the dataset should match the variable label described in ADaM. When creating a new domain Variable Labels could be adjusted as appropriate. Currently this rule runs if a variable has a Label in the OpenCDISC ADaM metadata. | Metadata | Error |
|--------|----|---|--|----------|-------|

Figure 9.b

The error in figure 9.b is very likely to occur while making use of the labels in ADaM Implementation Guide 1.1. Some labels have been updated as they were previously incorrect and misleading so some of them changed. For example, between the ADaM Implementation Guide versions 1.0 and 1.1, the variable SRCDOM got a slightly updated label; from *Source Domain* in 1.0 to *Source Data* in 1.1. While it is a minor difference, using the 1.1 label results in a non-compliant ADaM 1.0 database.

| | | | | | |
|--------|--|---------------------|--|-------------|-------|
| DD0055 | | Invalid Class value | The Class attribute must have a value of 'SPECIAL PURPOSE', 'FINDINGS', 'EVENTS', 'INTERVENTIONS', 'TRIAL DESIGN', or 'RELATIONSHIP' for SDTM and SEND data and 'SUBJECT LEVEL ANALYSIS DATASET', 'BASIC DATA STRUCTURE', or 'ADAM OTHER' for ADaM data. | Terminology | Error |
|--------|--|---------------------|--|-------------|-------|

Figure 9.c

Finally, in Figure 9.c we have an error that occurred when the define.xml is validated separately in P21. While the CDISC codelist General Observation Class includes values such as IADSL and OCCDS, the models underlying these values clearly state that they were designed to be used in conjunction with ADaM Implementation Guide 1.1, and therefore they are not compliant with ADaM Implementation Guide 1.0.

It is a fair assumption that implementation based on ADaM Implementation Guide 1.1 or ADaM Structure for Occurrence Data 1.0 will not impair the quality of the derivations, and traceability, in fact the contrary is likely to happen as these guidelines, in our opinion, are improvements compared to their predecessors (ADaM IG 1.0 and ADAE 1.0). However, to avert unnecessary risk any sponsor planning to submit data (single study or as an integrated analysis) should stay

⁷ FDA's Study Data Technical Conformance Guide Version 3.3, PMDA's Technical Conformance Guide on Electronic Study Data Submissions July 2015

⁸ Ibid.

⁹ FDA Data Standards Catalog v. 4.5.2

(<https://www.fda.gov/ForIndustry/DataStandards/StudyDataStandards/ucm2005545.htm>) and PMDA's Data Standards Catalog (2017-03-03) (<https://www.pmda.go.jp/english/review-services/reviews/advanced-efforts/0002.html>)

PhUSE 2017

compliant to the standards supported in the FDA's or PMDA's standards catalogs. We still encourage anyone designing ADaM datasets to consult the most recent published implementation guides for more accurate examples and descriptions as long as the supported standards are followed.

CONCLUSION

Technical tools and CDISC-savvy programmers are undisputedly a necessity for any CRO or sponsor organization implementing regulatory standards. Without ignoring the contribution of such technical solutions, our aim here was to illustrate the benefit that a shift of focus towards the underlying production processes can provide in order to reduce the risk of missed timelines, budget overburn, and countless hour of re-work. The suggestions we provided for avoiding the four main pitfall examples we described are surely not an exhaustive list of possible process enhancements, but we are convinced that these strategies, in conjunction with the available technical tools and good CDISC expertise, will maximize efficiency and quality on any project.

PhUSE 2017

REFERENCES

Analysis Data Model Implementation Guide Version 1.0, CDISC
Analysis Data Model Implementation Guide Version 1.1, CDISC
ADaM Structure for Occurrence Data (OCCDS) Version 1.0 CDISC
Analysis Data Reviewer's Guide Completion Guidelines Version 1.1 2015-01-26 PhUSE
(http://www.phusewiki.org/wiki/index.php?title=Analysis_Data_Reviewer%27s_Guide)
External guidance on the implementation of the European Medicines Agency policy on the publication of clinical data for medicinal products for human use, version 1.2 published 12 April 2017, EMA
(http://www.ema.europa.eu/docs/en_GB/document_library/Regulatory_and_procedural_guideline/2017/04/WC500225880.pdf)
Study Data Reviewer's Guide Completion Guidelines Version 1.2 2015-01-26 PhUSE
(http://www.phusewiki.org/wiki/index.php?title=Study_Data_Reviewer%27s_Guide)
FDA Study Technical Conformance Guide Version 3.3 FDA
(<https://www.fda.gov/downloads/forindustry/datastandards/studydatastandards/ucm384744.pdf>)
PMDA Technical Conformance Guide on Electronic Study Data Submissions July 2015 PMDA
(<https://www.pmda.go.jp/files/000206449.pdf>)

CONTACT INFORMATION

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

Hannes Engberg Raeder
PRA Health Sciences
Gottlieb-Daimler-Strasse 10 68165 Mannheim, Germany
engbergraederhannes@prahs.com
prahs.com

Michael Reich
PRA Health Sciences
Gottlieb-Daimler-Strasse 10 68165 Mannheim, Germany
reichmichael@prahs.com
prahs.com