PharmaSUG 2017 - Paper AD19

SAS Application to Automate a Comprehensive Review of DEFINE and All of its Components

Walter Hufford, Vincent Guo, and Mijun Hu, Novartis Pharmaceuticals Corporation

ABSTRACT

The DEFINE is a large electronic document comprised of many different but interrelated components such as annotated Case Report Form, Data Reviewer's Guide and metadata. Further complicating the situation is that most electronic submissions contain several studies and an SCS and an SCE, each of which requires their own DEFINE. Reviewing the DEFINE to ensure consistency, accuracy and completeness within a single DEFINE as well as across DEFINEs is both time consuming and resource intensive (and often mind-numbing if you have a large submission with many DEFINEs to review). Automating review of the DEFINE can be achieved with a few simple, easy to develop, SAS® macros. The result is a much quicker review requiring substantially less resources. In addition, findings are noted in a standardized manner allowing for quicker issue resolution and tracking. We will describe our DEFINE review tool in detail and provide code snippets from our macros which should allow you to implement a similar tool with little effort.

INTRODUCTION

The DEFINE package is a large electronic document comprised of many different but interrelated components with the define.xml acting as a road map. Embedded hyperlinks allow reviewers to easily navigate between components (annotated Case Report Form, Data Reviewer's Guide, SAS transport (XPT) files, and metadata) with the goal of understanding how the data is collected or derived for the analysis purpose. In order to achieve the goal, the DEFINE must be accurate, complete and consistent both within and between the components. It is a massive undertaking to review all the components to ensure accuracy, completeness and consistency once the DEFINE package is created, especially when it is done manually.

Automating the manual review process can be achieved once you (1) are familiar with the DEFINE and all of its distinct, interrelated components and sections, (2) fully understand the scope of what a thorough review of the DEFINE entails, and (3) construct a data structure capable of consolidating disparate metadata from each of the DEFINE components.

Automation of the DEFINE review eliminates incomplete and inconsistent findings and spares you from lengthy, tedious, and repeated manual reviews. As a result, programming and statistical resources are able to focus on tasks which require a higher level of functional expertise and knowledge.

DEFINE COMPONENTS AND STRUCTURE

SDTM and ADaM DEFINES share the same components and structure with the exception of the annotated Case Report Form (SDTM only) and the Analysis Results Metadata (ADaM only). At first glance the DEFINE appears complex as illustrated in Figure 1. However, when broken down into its distinct components and sections, the DEFINE is actually quite straightforward and somewhat similar to a SAS PROC CONTENTS on steroids.

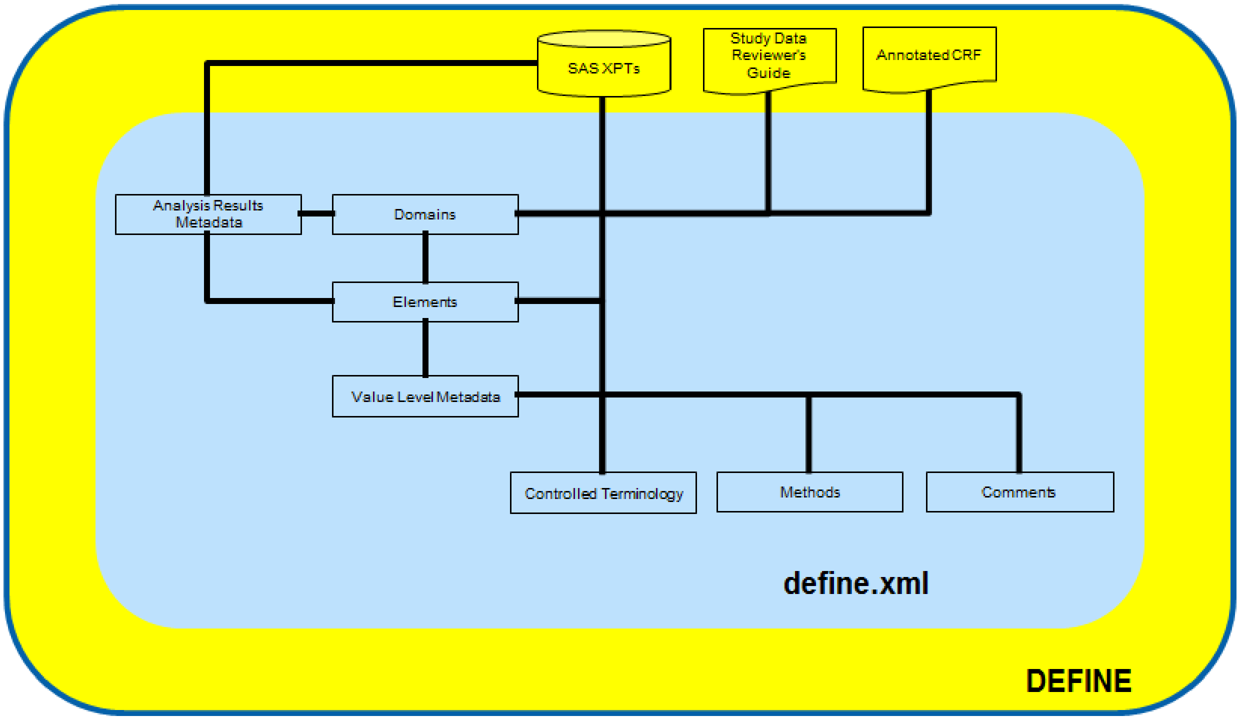


Figure 1. Interrelated DEFINE Components and Structures

COMMON SDTM AND ADAM DEFINE COMPONENTS AND STRUCTURE

The following DEFINE components and sections are common across SDTM and ADaM.

The Data Reviewer's Guide is an external (to the define.xml) PDF file accessed thru the define.xml. It is the first document a regulatory reviewer reads before starting their review of the data. The purpose of this document is to provide the reviewer additional information beyond that provided in the body of the define.xml itself. For example, it describes the data standard version and controlled terminology used, additions to extensible controlled terminology, additional categorization and grouping variables, complex derivation rules, and usage of critical variables in the analysis.

The first section of the define.xml contains domain-level metadata. It is the least granular section of the define.xml and resembles a table of contents. All data domains and the attributes for each (e.g., class, structure, keys) are listed in this section. Hyperlinks exist to corresponding data domain SAS transport (XPT) files and the element-level metadata section.

The SAS transport (XPT) files are external (to the define.xml) files accessed thru hyperlinks found in the domain and element-level metadata sections of the define.xml.

The element-level metadata section follows the domain-level metadata. Domain elements and their attributes (e.g., type, controlled terminology, derivation) are listed in this section. Hyperlinks exist to corresponding value-level metadata, controlled terminology, methods and comments sections, the annotated Case Report Form (SDTM only) and the Data Reviewer's Guide.

The value-level metadata (VLM) section follows the element-level metadata section. It is needed when values and/or attributes of an element are derived and defined differently under certain conditions involving one or more other elements. For example, some SDTM and ADaM element such as xxORRES and AVAL may have different derivations dependent upon another element (or elements) value (e.g., xxTESTCD, PARAMCD). Hyperlinks exist to corresponding element-level metadata, controlled terminology, methods and comments sections, the annotated Case Report Form (SDTM only), and the Data Reviewer's Guide.

The controlled terminology (CT) metadata section follows the value-level metadata section. Controlled terminology values used to describe the data within the define.xml are listed in this section.

The methods section follows the controlled terminology section. Derivations displayed in the element-level and value-level metadata sections are listed in this section.

The comments section follows the methods section. Comments displayed in the element-level and value-level metadata sections are listed in this section.

SDTM SPECIFIC DEFINE CONPONENTS

The annotated Case Report Form is an external (to the define.xml) PDF file accessed via the SDTM define.xml. It contains all of the unique pages from the case report form (CRF) used in the study. Each unique page contains annotations identifying SDTM elements collected on that page.

ADAM SPECIFIC DEFINE COMPONENTS

The Analysis Results Metadata (ARM) is the first section (prior to the domain-level metadata section) of an ADaM define.xml. It facilitates traceability between ADaM datasets and a set of key analysis results. Hyperlinks exist to corresponding element-level metadata, the Data Reviewer’s Guide, and the output (e.g., table, listing or graph).

REVIEW CHALLENGES

A single DEFINE contains hundreds of data elements and values each of which needs to be reviewed for accuracy, consistency, and traceability. Each of these data points must be re-verified each time a new draft is generated even if only a single data point is updated. As Figure 2 illustrates, most electronic submissions are comprised of more than one study, an SCS and an SCE, further increasing the magnitude of the review. Often some issues identified during the review of the DEFINE for one study were overlooked during the review of the DEFINE for another study leading to an update and re-review of the previously approved DEFINE.

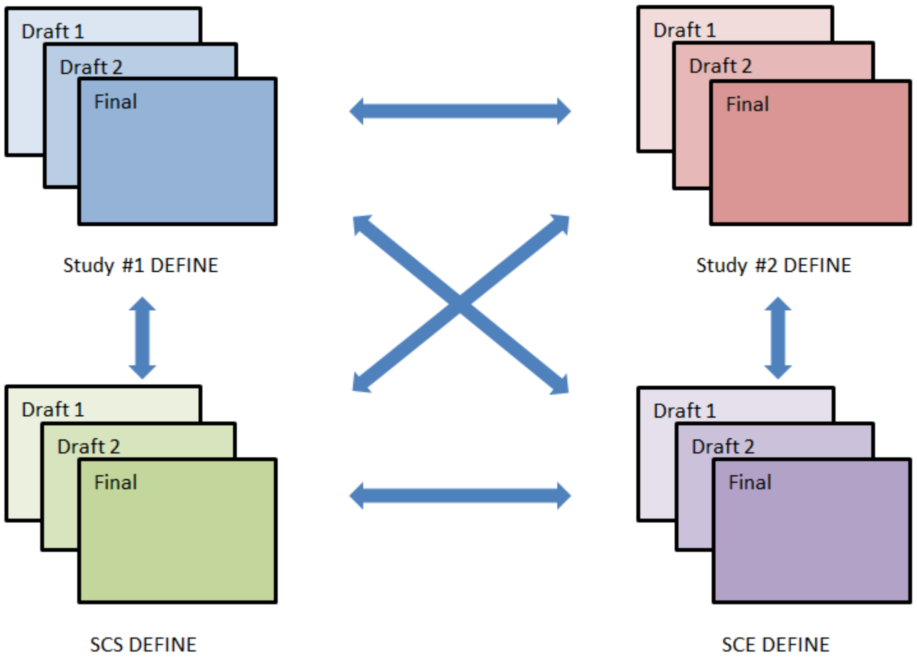


Figure 2. Interrelated DEFINEs within a Submission

In addition, the distinct, interrelated components of the DEFINE described in the previous section introduce more challenges that need to be addressed. Examples of these challenges are:

1. Consistency and traceability within define.xml
   - Elements specified as derived must have comment fields populated with a derivation
   - Elements referred to in the derivation must be contained within the submitted domains for traceability
2. define.xml must be consistent with
   - Input (i.e., specifications) to the DEFINE (e.g., attribute and derivations, VLM, computational methods)
   - XPT files (e.g., XPT files must be sorted by same KEYS listed in the define.xml, CT values in define.xml must be the same as XPT element values)
3. CRF annotations must be consistent with
   - XPT files (e.g., each annotated domain must have a corresponding XPT file)
   - the define.xml (e.g. elements in define.xml with origin as CRF must appear in the annotated Case Report Form, annotated Case Report Form elements must appear in define.xml with origin as CRF)
4. Data Reviewer's Guide must be consistent with
   - XPT files (e.g., CAT and SCAT elements and values, supplemental qualifier and values, extensible controlled terminology)
   - define.xml (e.g., data structure, version of dictionaries used)

Conducting manual reviews of so many data points is time-consuming, resource-intensive, sometimes painful, and often incomplete. The next sections demonstrate how automation can overcome these challenges to achieve maximum efficiency and effectiveness while conducting a comprehensive review of the DEFINE.

METHODS

The first step in creating an automated DEFINE review tool is to convert the different components into a single .sas7bdat format. Different conversion methods are employed dependent upon the file type. Simple, straightforward SAS DATASTEP code is used to conduct the QC once all components are formatted consistently. The following steps, as illustrated in Figure 3, provide details of (1) the conversion process, (2) the QC code implementation, and (3) creation of the issue identification and resolution file.

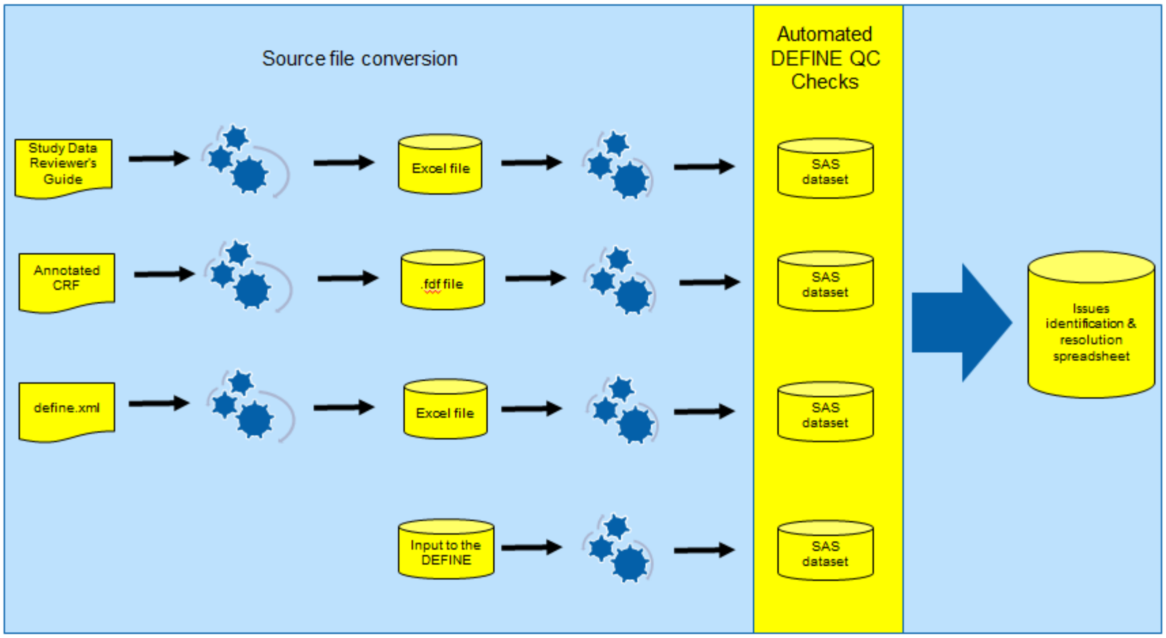


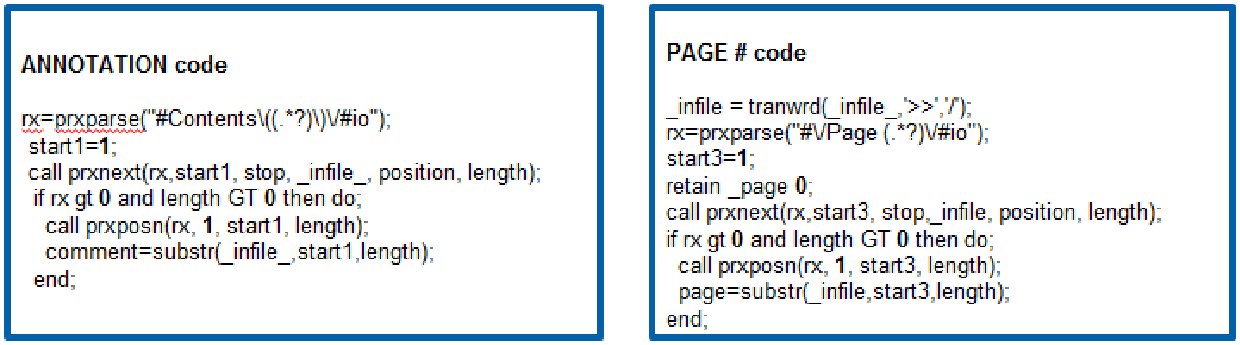
Figure 3. Automated DEFINE Review Tool Overview

STEP 1: DATA REVIEWER’S GUIDE CONVERSION

The Data Reviewer’s Guide is written in WORD and converted to PDF format for inclusion in the DEFINE. A simple Visual Basic macro written and installed in an Excel file reads the WORD document and creates an Excel file containing a tab for each table within the WORD document. The Excel file is easily converted to the common .sas7bdat format using PROC IMPORT.

STEP 2: ANNOTATED CASE REPORT FORM CONVERSION

The annotated Case Report Form is created from an Adobe .fdf file (see Hufford 2014). The .fdf file is a structured flat file and is easily converted to the common .sas7bdat format using simple SAS DATASTEP code (e.g., INFILE and INPUT statements and some SAS regular expression code used to scan entire records for annotations and their attributes). Examples of SAS regular expression code used to obtain annotation and page number are illustrated in Figure 4.



ANNOTATION code

```
rx=prxparse("#Contents\((.*?)\)\/#io");
 start1=1;
 call prxnext(rx,start1, stop, _infile_, position, length);
  if rx gt 0 and length GT 0 then do;
    call prxposn(rx, 1, start1, length);
    comment=substr(_infile_,start1,length);
  end;
```

PAGE # code

```
_infile = tranwrd(_infile_,'>>','/');
rx=prxparse("#\/Page (.*?)\/#io");
start3=1;
retain _page 0;
call prxnext(rx,start3, stop,_infile, position, length);
if rx gt 0 and length GT 0 then do;
  call prxposn(rx, 1, start3, length);
  page=substr(_infile,start3,length);
end;
```

Figure 4. Annotated Case Report Form Code Snippets

STEP 3: DEFINE.XML CONVERSION

The define.xml file is opened in Excel (without applying a stylesheet) and saved as an .xls file format. The Excel file is then easily converted to the common .sas7bdat format using PROC IMPORT and simple SAS DATASTEP code once you have a full understanding of the relationships between the sections within the define.xml. A separate .sas7bdat file is created for each define.xml section (e.g., domain-level metadata, element-level metadata, VLM, controlled terminology, methods, and comments).

STEP 4: INPUT (I.E., SPECIFICATIONS) TO THE DEFINE

Specifications in an Excel format (see Step 1 above for conversion from WORD to Excel format) are easily converted to the common .sas7bdat format using PROC IMPORT and simple SAS DATASTEP code. A separate .sas7bdat is created for each specification section in the same manner as is done during the define.xml conversion.

STEP 5: CREATE AUTOMATED DEFINE REVIEW TOOL

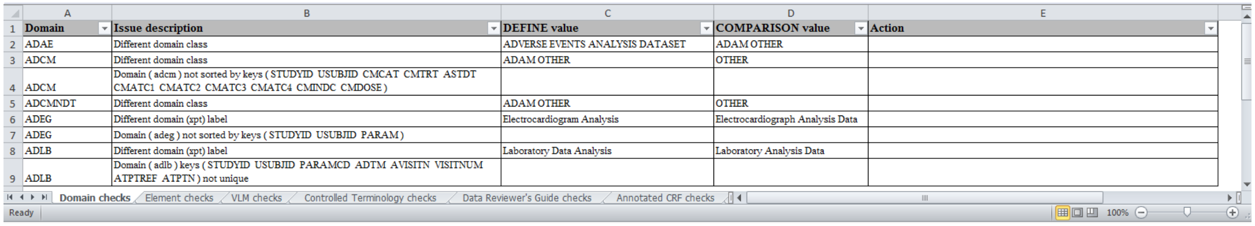
The creation of the automated DEFINE review tool is a relatively straightforward task once all conversions to the single .sas7bdat format are complete. Simple SAS DATASTEP code consisting of PROC SORT and MERGE statements with IN operators is used to identify QC issues.

As discussed in the Review Challenges section, hundreds of individual data points are checked for accuracy, consistency and traceability. In addition to these consistency checks (i.e., within DEFINE and between DEFINE and specifications), other checks which are not currently covered by Pinnacle 21 software are also conducted (e.g., when ORIGIN=DERIVED but no derivation is provided).

Once both SDTM and ADaM defines are produced, cross-define consistency checks are implemented to ensure traceability (e.g., ADaM define ADSL element comment refers to an SDTM element which does not exist in the SDTM define).

STEP 6: CREATE ISSUES IDENTIFICATION AND RESOLUTION SPREADSHEET

Finally, all QC issues are consolidated into a single Excel file as illustrated in Figure 5. Separate tabs are used to store related issues (e.g., domain-level, element-level, controlled terminology, VLM, Data Reviewer's Guide, annotated Case Report Form [SDTM only], ARM [ADaM only]). Standard columns are created in the issue file including dataset, element, issue description, define value, comparison value (e.g., specifications, annotated Case Report Form) and action. Traffic lighting is added to assist with the prioritization of issue resolution. For example, if the only difference between a specification derivation and a define.xml derivation is due to spacing, then the row is highlighted in yellow. The action column is left blank when the issue file is created. A meeting is held to discuss an appropriate action for each issue (e.g., update DEFINE derivation to match specification).



	A	B	C	D	E
1	Domain	Issue description	DEFINE value	COMPARISON value	Action
2	ADAE	Different domain class	ADVERSE EVENTS ANALYSIS DATASET	ADAM OTHER	
3	ADCM	Different domain class	ADAM OTHER	OTHER	
4	ADCM	Domain (adcm) not sorted by keys (STUDYID USUBJID CMCAT CMTRT ASTDT CMATC1 CMATC2 CMATC3 CMATC4 CMINDC CMDOSE)			
5	ADCMNDT	Different domain class	ADAM OTHER	OTHER	
6	ADEG	Different domain (xpt) label	Electrocardiogram Analysis	Electrocardiograph Analysis Data	
7	ADEG	Domain (adeg) not sorted by keys (STUDYID USUBJID PARAM)			
8	ADLB	Different domain (xpt) label	Laboratory Data Analysis	Laboratory Analysis Data	
9	ADLB	Domain (adlb) keys (STUDYID USUBJID PARAMCD ADTM AVISITN VISITNUM ATPTREF ATPTN) not unique			

Figure 5. Issues Identification and Resolution Spreadsheet

RESULTS/DISCUSION

Prior to implementing the automated DEFINE review tool, a review of the DEFINE was a manual task which took several weeks to complete. Each reviewer employed their own arsenal of strategies and techniques. Some common manual review strategies and techniques include:

- line-by-line visual compare (e.g., side-by-side comparison between specifications, components, iterations, and DEFINEs) of all components and data points
- basic SAS code checks (e.g., PROC FREQ of XPT file values) to ensure the define.xml, annotated Case Report Form, SDRG/ADRG are consistent with XPT files
- manual documentation of findings in various formats (e.g., email, Excel spreadsheet) with varying levels of detail

These manual reviews were incredibly inefficient. Not only did they take a lot of time to complete, two different reviewers might be reviewing the same components in the same manner using the same criteria.

In addition, review criteria are not always well documented during the manual process. Even when QC checklists are available, reviewers may deviate from them, so it may not be guaranteed that a comprehensive review is conducted. A reviewer might skip the review of a section assuming the other reviewer would check it. All too often the review of the DEFINE for a particular study may identify issues which were not identified during the previously concluded review of the DEFINE for another study.

After implementing the automated DEFINE review tool, we realized a significant savings in the required resources (e.g., time and FTEs) to conduct a review of the DEFINE. On average, we estimate automation realized about a 2/3 time reduction as illustrated in Figure 6. The savings for one DEFINE that required three iterations of review is 14 days. These time savings are multiplied when considering most submissions consist of multiple studies, an SCS and an SCE, each of which requires its own DEFINE. The savings remain significant even after considering the initial creation of the automated DEFINE review tool which requires about a two week full-time investment by a single programmer and the periodic maintenance including upgrades (e.g., define 1.0 to define 2.0), addition of new review criteria, and bug fixes. The more the tool is used, the more savings are achieved.

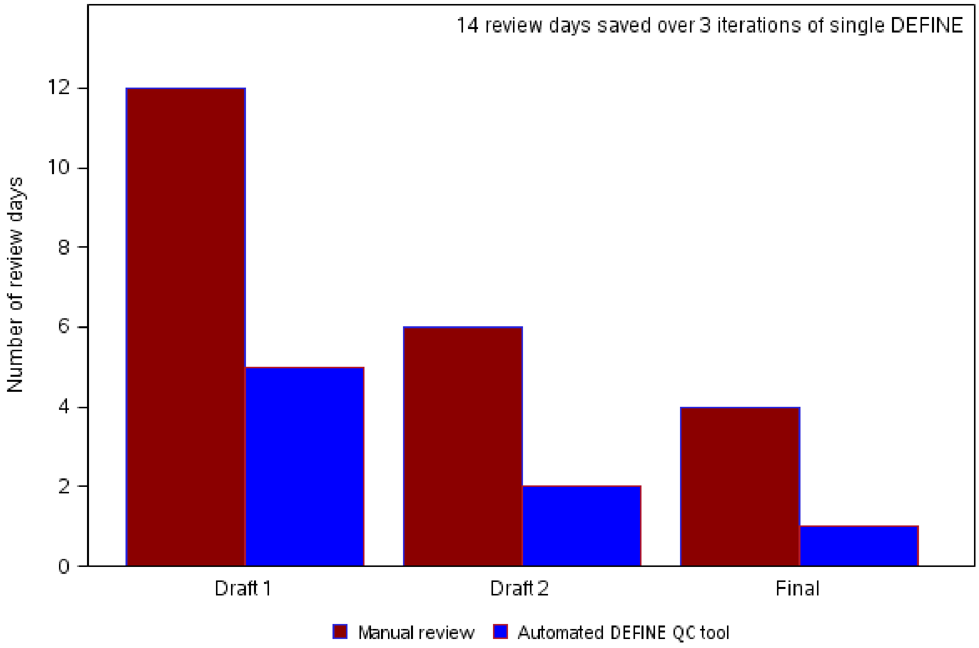


Figure 6. Average Time Savings After Implementing Automated DEFINE Review Tool

Factors such as level of experience, fatigue, and time constraints which lead to inconsistent and incomplete manual reviews of the DEFINE are eliminated during automation. The same set of pre-defined review criteria are implemented each time in a systematic manner, across the entire DEFINE. In the past, following a manual review, reviewers would meet to discuss, agree to and consolidate their independent findings prior to taking corrective action. The automated DEFINE review tool assigns a pre-approved message to each finding thus reducing the time required to meet and understand each finding before taking corrective action.

It is also worth noting the pivotal role that well-conceived project-level input specifications (i.e., single specification file which applies to all studies contained in the submission) play. These specifications must be accurate and concise, and consolidate project-level data point attributes (e.g., AE.AESER should have all the same attributes across all studies within the submission) while allowing for the existence of study-specific deviations and additions where necessary (e.g., only a single study within the submission collects x-ray data). They can also be used to drive the code used to create the tabulation and analysis datasets and produce and QC the DEFINE. Thus, it is important to consider all data attributes required for dataset and DEFINE when designing your specifications. In turn, automated DEFINE review tool findings sometimes contribute to improving the quality of your specifications (e.g., an ADaM DEFINE comment refers to a SDTM element which does not exist).

CONCLUSION

The DEFINE is a large, complex, interrelated electronic document. A comprehensive review of the DEFINE is a massive undertaking especially when it is done manually. Once you fully understanding each of the interrelated DEFINE components and construct a data structure capable of consolidating disparate metadata from each, automating review of the DEFINE can be easily achieved via simple SAS DATASTEP code. Automation eliminates incomplete and inconsistent findings, reduces the burden on programming and statistical resources, and greatly improves the quality of the review of the DEFINE.

REFERENCES

Walter Hufford, “Automating Production of the blankcrf”. URL: http://www.pharmasug.org/2014-proceedings.html

CONTACT INFORMATION

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

Walter Hufford
Novartis Pharmaceuticals Corporation
Walter.Hufford@Novartis.com

Vincent Guo
Novartis Pharmaceuticals Corporation
Vincent.Guo@Novartis.com

Mijun Hu
Novartis Pharmaceuticals Corporation
Mijun.Hu@Novartis.com