

# Generating Data Definition Domain Documentation DEFINE.XML

MXI, Meta-Xceed, Inc.



**Sy Truong**  
System Architect  
President of MXI, Meta-Xceed, Inc.



November 5, 2008

## **MXI Profile**

- **Clinical Analytical Software & Services for Pharmaceutical and Biotech since 1997**
- **MXI Specializes in:**
  - CFR Part 11 Data Standards and CDISC
  - Electronic Submission / FDA Compliance
  - Validation of Statistical Programming and Biostatistics Systems
- **Headquarters: San Jose USA**  
**Software Development Center: Vietnam**

# COURSE OUTLINE

- XML Primer
- XML Structure by Example
- Data Definition Documentation Background
- Define.XML Structure



# Define.xml and CDISC Milestones

- **Study Data Tabulation Model (SDTM)**
  - Referenced in FDA Guidance as of 21 July 04
- **Define.xml for submission of SDTM metadata**
  - Referenced in FDA Guidance as of 18 March 05
- **ODM for submission of original CRF data**
  - Pilot project: Federal Register announcement, March 13, 2007
- **SDTM Endorse by FDA May 2008**

# XML Key Advantages

- Vendor Neutral
- Platform Independent, Flexible
- Designed for Interchange of data between Heterogeneous Systems
- Files can be Automatically Checked for Syntactical, Structural and Semantic Correctness (with XML schemas)

# Define.xml Usage

- Data Interchange or Transfer
- Data Archival
- Data Project Management
- Data Standards
- Data Integrity Review



# COURSE OUTLINE

- XML Primer
- XML Structure by Example
- Data Definition Documentation Background
- Define.XML Structure

# XML from W3C

- Mission of W3C is to develop interoperable technologies to lead the Web to its full potential
- Web Address  
<http://www.w3.org>



The screenshot shows the W3C (World Wide Web Consortium) homepage. At the top, the W3C logo is displayed with the tagline "Leading the Web to Its Full Potential...". Below this, a navigation bar contains links for "Activities", "Technical Reports", "Site Index", "New Visitors", "About W3C", "Join W3C", and "Contact W3C". A paragraph of text describes the W3C's mission to develop interoperable technologies. Below the text, there are three main sections: "W3C A to Z" with a list of links (Accessibility, Amaya, Annotea, Binary XML, CC/PP, Compound Document Formats, CSS, CSS Validator, Device Independence, DOM), "News" with a headline "'Architecture of the World Wide Web, Volume One' is a W3C Recommendation" and a sub-headline "2004-12-15: The World Wide Web Consortium today released Architecture of the World Wide Web, Volume One as a W3C Recommendation. The Web uses relatively simple technologies with sufficient...", and "Search" with a Google search bar and links for "Search W3C" and "Search W3C Mailing Lists". A "Members" section is partially visible at the bottom right, showing "The Hebrew University of..."



# XML Background

- XML (Extensible Markup Language) is a markup language for documents containing structured information
- Derived from SGML – as is HTML
- A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents
- The focus of XML is on content and structure, not presentation
- In contrast to HTML where focus is on presentation

# XML Tags

- **Tags**
  - Enclosed within < and > characters
- **Start Tag**  
<TAG>
- **End Tag**  
</TAG>
- **Can Combine**  
<TAG/>

# XML Elements

- Everything between the start of a start-tag to the end of the end-tag

`<TAG>This is the tag content</TAG>`

- Element name: TAG
- Element content: This is the tag content



# XML Element Hierarchy

**<Name>**

**<FirstName>Sy</FirstName>**

**</Name>**

- **FirstName** is a child of **Name**
- **Name** is the parent of **FirstName**

# Elements Siblings

<Name>

<FirstName>Sy</FirstName>

<LastName>Truong</LastName>

</Name>

# XML Comments

`<!-- This is a comment -->`

`<!-- A another comment  
over a line -->`



# Elements Must Not Overlap

- The Following is Invalid

<ContactList>

<Name>

<FirstName>Fred</FirstName>

<LastName>Smith

</Name> </LastName>

</ContactList>

# Element Attributes

- **Start-tags can also have attributes**
  - `<FirstName nickname="Cat">Catherine</FirstName>`
  - Attribute name: `nickname`
  - Attribute value: `Cat`
- **Can have multiple attributes**
  - `<TAG attr1="1" attr2="2">Content Here</TAG>`

## COURSE OUTLINE

- XML Primer
- **XML Structure by Example**
- Data Definition Documentation Background
- Define.XML Structure



# XML CD Example



## CD Collection

1.CD

1.Artist

2.Title

3.Track 1

4.Track 2

5.Track 3

6....

2.CD

3.CD

4....

## CD Collection Root Element

```
<CDCollection>  
  <!-- Something Here -->  
</CDCollection>
```

## CD Child Elements

```
<CDCollection>  
  <CD>  
    <!-- Something Here -->  
  </CD>  
  <CD>  
    <!-- Something Here -->  
  </CD>  
  <!-- More CDs here -->  
</CDCollection>
```



## CD Artist, Title and Track Elements

```
<CDCollection>
  <CD>
    <Artist>Artist's Name</Artist>
    <Title>Title of CD</Title>
    <Track>Track 1 title</Track>
    <Track>Track 2 title</Track>
    <!-- More Tracks here -->
  </CD>
  <!-- More CDs here -->
</CDCollection>
```

# Dark Side of the Moon CD Example

```
■ CDCollection> Structure
  ■ CD TotalTime="45.02">
    ■ Artist>Pink Floyd</Artist>
    ■ Title>Dark Side of the Moon</Title>
      <Track Label="1a">Speak To Me</Track> Element
      <Track Label="1b">Breathe</Track>
      <Track Label="2">On the Run</Track>
      <Track Label="3" Attribute ck>
      <Track Label="4">The Great Gig in the Sky</Track>
      <Track Label="5">Money</Track>
      <Track Label="6">Us and Them</Track>
      <Track Label="7">Any Colour You Like</Track>
      <Track Label="8">Brain Damage</Track>
      <Track Label="9">Eclipse</Track>
    </CD>
  </CDCollection>
```

# XML Schema

- Schema used for defining structure, content and semantics of XML documents
- Like Grammar, Language for expressing shared vocabularies
- Evolved from document type definitions (DTD)



# XML Schema Provides

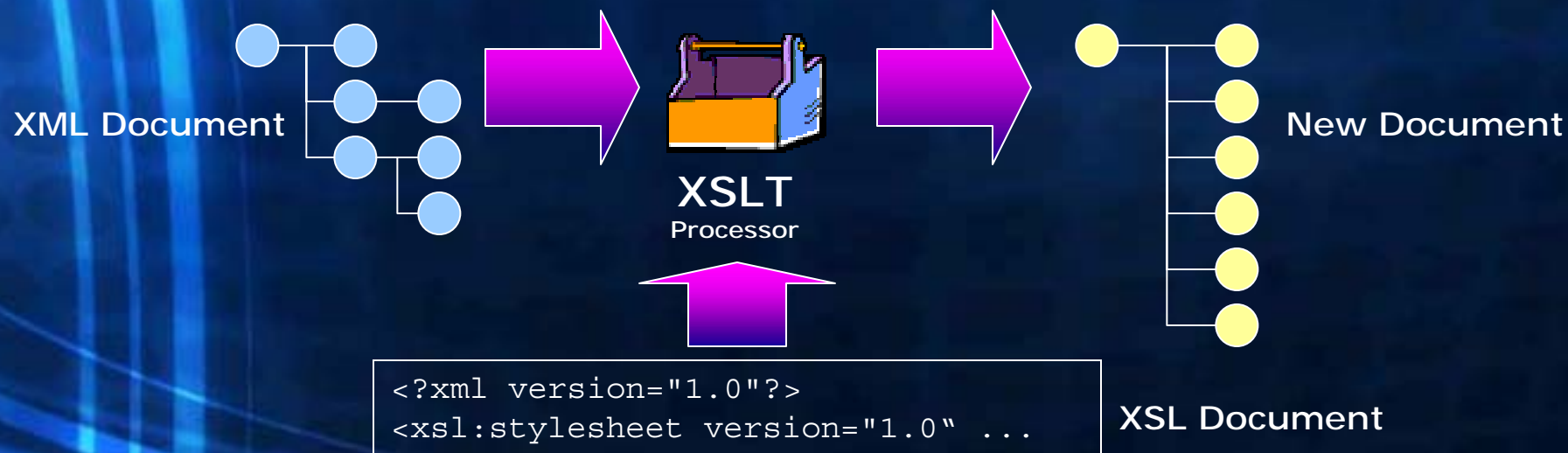
- Define XML Namespaces
- Define datatypes, elements, attributes
- Define relationships between elements, repeatability
- Provide an outline for an XML instance document
- Support standardization of XML documents
- Enable validation of XML documents

# Schema Examples

- DTD
  - Document Type Definition
  - Grammar for XML
  - CD.DTD
- XSD
  - XML Schema Definition
  - CD.XSD
  - Newer than DTD
  - Used to Validate XML

# XML Transformation

- XSL – Extensible Stylesheet Language
- Used to transform an XML document
- Requires a tool known as XSLT processor
- Focuses on presentation while XML focuses on content and structure

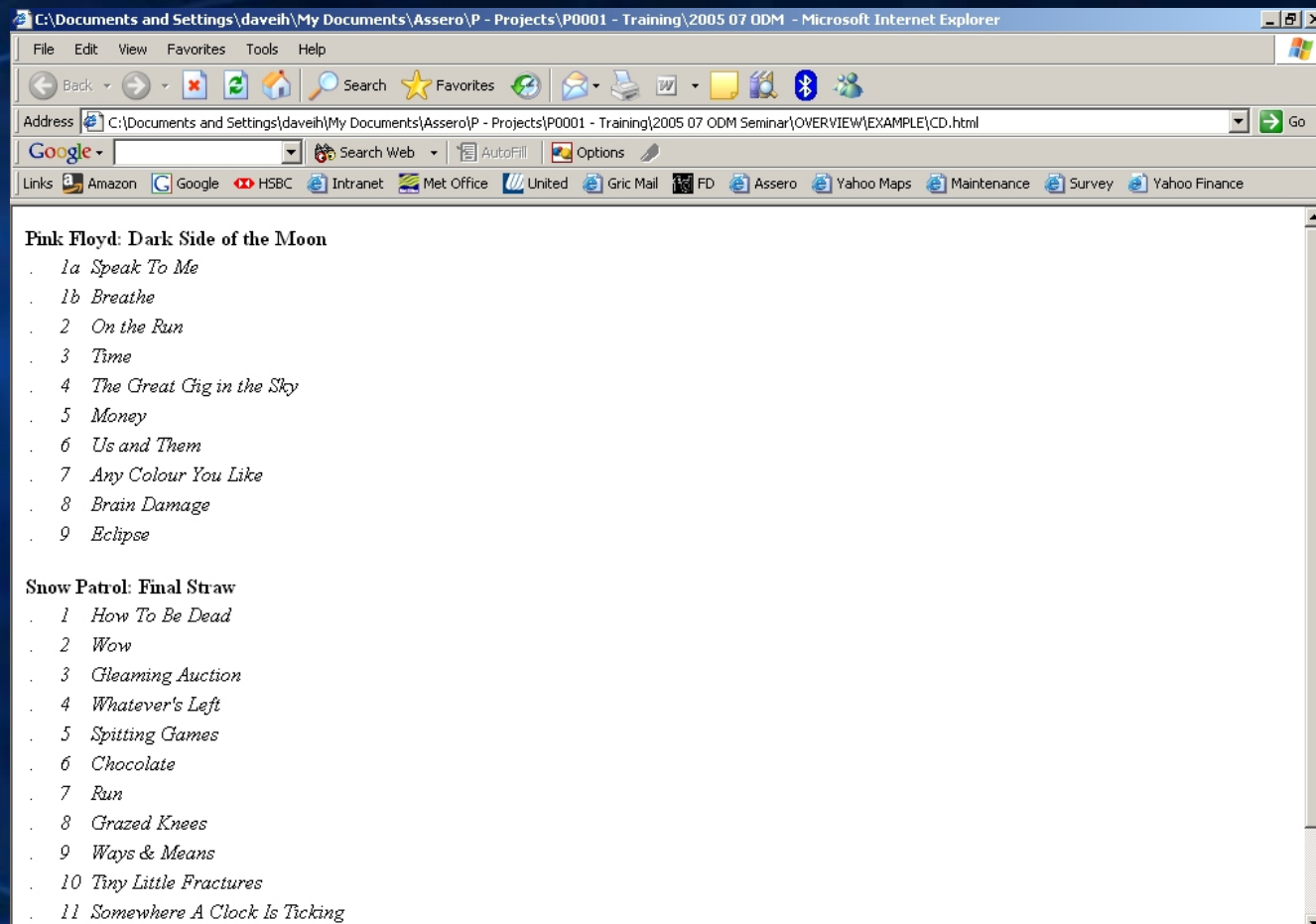




# CD Example Completed

- **Schema**
  - CD.XSD
- **Data**
  - CD.XML
- **Transformations**
  - CD.XSL
- **Results**
  - CD.HTML

# CD Example Output View



## Applying XSL

- XSL may be use stand-alone to convert DEFINE.XML files into a variety of formats, including HTML, CSV and SAS
- XSL may be used in tandem with other technologies for more sophisticated applications

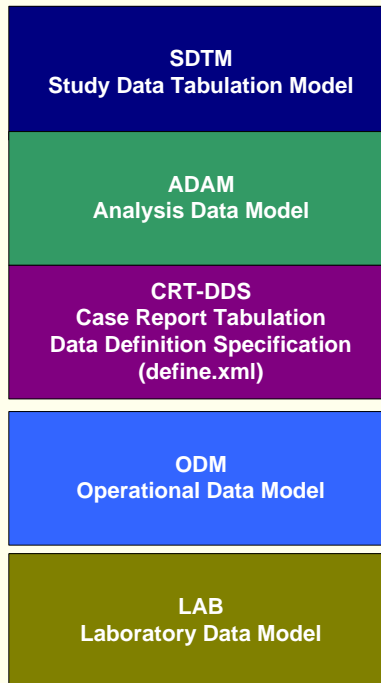


## COURSE OUTLINE

- XML Primer
- XML Structure by Example - XML CD Example
- **Data Definition Documentation Background**
- Define.XML Structure

# CDISC Related Models

## CDISC Data Models



# Standard Guidelines




- <http://cdisc.org/standards/index.html>

## Standards

[CDISC Technical Roadmap](#)

Quick Links to Models:

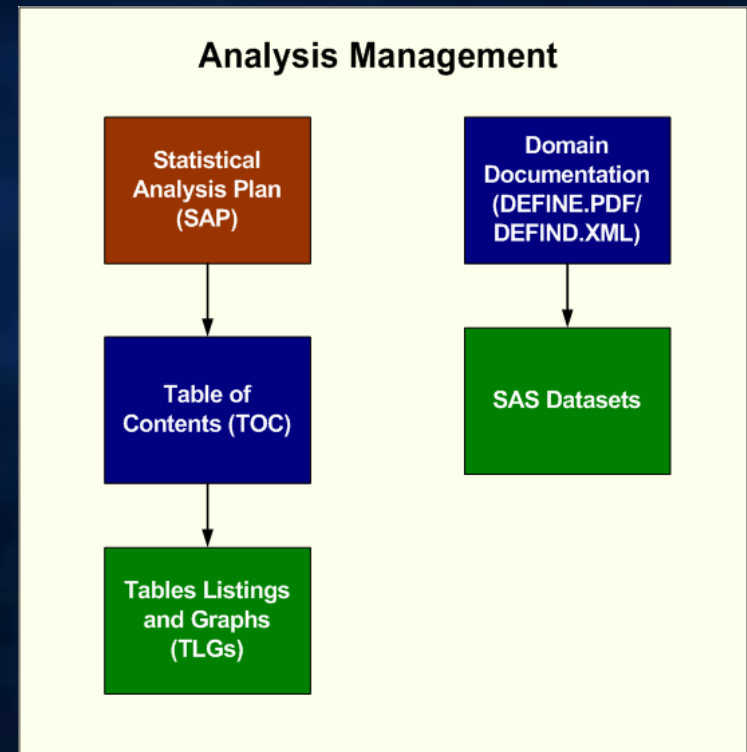
### Standards in Production

Submission Data Standards Team	(SDTM IG V3.1.1)
	(SDTM V1.1)
	(SDTM IG V3.1) 
	WebSDM edit checks for (SDTM 3.1.1)
Operational Data Modeling Team	(ODM V1.3)
	(ODM V1.2.1)
	See also (eSDI Document)
Analysis Dataset Model Team	(ADaM 2.0) 
Laboratory Standards Team	(LAB)
Standard for Exchange of Non-clinical Data	(SEND V2.3)
Case Report Tabulation Data Definition Specification (define .xml)	(CRT-DDS V1.0) 
Terminology	(Terminology)



# Data Definition Introduction

- DEFINE.PDF/XML is like a TOC (Table of Contents) or SAP (Statistical Analysis Plan) Guiding Document
- Effective Tool for Managing SAS Datasets



# DOCUMENTATION TASKS

- **Tasks Needed to be Performed for Documentation:**
  - Organizing Datasets to be Documented
  - Documenting Variable Attributes
  - Decoding Format Codes
  - Determining the Origins or Source of the Data
  - Documenting Comments
  - Ways of Automating the Capture and Presentation of the Metadata

# WAYS OF USING DOCUMENTATION

- Project Management - Organize SAS Data and Status of Development
- Data Standards - Maintain Data Structure Standards
- Data Integrity and Review - Data Quality Review and Audits



## QUIZ 1

- Which one of the following analysis and reporting concepts is analogous to data definition documentation?
- A) CDISC SDTM Data Model
- B) TOC, Table of Contents
- C) SAP, Statistical Analysis Plan
- D) B and C

## QUIZ 2

- Which of the following tasks is **not** a needed in performing documentation?
- A) Documenting Variable Attributes
- B) Decoding Format Codes
- C) Generating Table of Contents of Tables
- D) Determining Origins of Source Data
- E) Documenting Comments

## QUIZ 3

- Which of the following are **not** commonly used with data definition documentation?
  - A) Data Integrity Review
  - B) Mockup for Analysis Files
  - C) Data Standards
  - D) Project Management



# USED FOR PROJECT MANAGEMENT

- TOC – Table of Contents
- Start at the End
- Step 1: List all Datasets for Electronic Submission

Dataset Name	Location	Keys	Number of Variables	Number of Records	Comment
ae	ae.xpt	usubjid	9	20	
cm	cm.xpt	usubjid	4	35	
co	co.xpt	usubjid	15	42	
dm	dm.xpt	usubjid	7	20	
ds	ds.xpt	usubjid	5	20	
ex	ex.xpt	usubjid	7	20	
relrec	relrec.xpt	usubjid	9	251	
su	su.xpt	usubjid	7	80	
suppqual	suppqual.xpt	usubjid	13	2102	

# MANAGE WITH COMMENTS

- **Sample Comments:**
  - Initial Draft, June 16, 2005 by ST
  - New derivation for age and weight unit calculations were added, June 17, 2005 by SW
  - Verification testing confirming calculations completed, June 20, 2005, by JD
  - Demographic data set with additional derivations for age and weight, July 21, 2005, by ST
- **Final Comments used for FDA Reviewer**

Dataset Name	Location	Keys	Number of Variables	Number of Records	Comment
ae	ae.xpt	usubjid	9	20	
cm	cm.xpt	usubjid	4	35	
co	co.xpt	usubjid	15	42	
dm	dm.xpt	usubjid	7	20	
ds	ds.xpt	usubjid	5	20	
ex	ex.xpt	usubjid	7	20	
relrec	relrec.xpt	usubjid	9	251	
su	su.xpt	usubjid	7	80	
suppqual	suppqual.xpt	usubjid	13	2102	

# PROJECT MANAGEMENT STEP 2

- Step 2: Document Variables and Attributes

ae (ae.xpt)								
Variable Name	Type	Length	Variable Label	Format	Decode Formats	Origins	Role	Comment
aeacn	Character	100	Action Taken with Study Treatment	actfmt.	1 = None 2 = Dose Increase 3 = Dose Decrease	Derived		
aeendy	Numeric	8	Study Day of End of Event			Derived		
Aesrc	Character	100	Adverse Event Collection Source			Derived		
aestdtc	Character	100	Start Date/Time of Adverse Event			Derived		
aestdy	Numeric	8	Study Day of Start of Event			Derived		
aeterm	Character	100	Reported Term for the Adverse Event			Derived		
siteid	Character	100	Study Site Identifier			Derived		
studyid	Character	100	Study Identifier			Derived		
usubjid	Character	100	Unique Subject Identifier			Derived	Key	



# DOCUMENT VARIABLE ATTRIBUTES

- **Variable Attributes**
  - Variable Name, Type / Length, Variable Label, Format / Decode Formats, Origins, Role, Comments
- **Determine All Derived Variables Needed**
- **Incorporated into SAP**
- **Use Comments to Track Status**

## Exercise 1

- Open [define\\_project\\_managment.rtf](#)
- Edit Dataset Level Comments
- Edit AE Adverse Event Variable Comments

# APPLY TO DATA STANDARDS

- Internal Standards or External CDISC Standards
- Document Data Attributes as Target
- Verify Against Standards before Programming
- Step 1: Review the DEFINE.PDF/XML Documentation at Dataset Level Against Standards



# DATA STANDARD STEPS

- **Step 2: Verify Define.PDF/XML Documentation Against Variable Attributes**
- **Step 3: Develop SAS Programs. Reconcile Standards Against Created Data**
  - Use PROC CONTENTS as Visual Inspection
  - Or use %CDISC or %DIFFTEST Macros to Automate (shown in Exercise)

# STANDARDS APPROACH

- **Portability of Data between Studies**
  - Portable SAS Programs
  - Easier for Users to Learn
- **Applied to Internal Standards or External CDISC Standards**
- **Documentation Change Control of Standards between Versions**

# DATA INTEGRITY REVIEW

- Imagine you are an FDA Reviewer
- Step 1: Verify DEFINE.PDF/XML Hyperlinks
- Step 2: Verify Keys Fields
  - Keys Listed First
  - Data is Sorted by Keys



# DATA INTEGRITY STEPS

- Step 3: Verify all Decoded Formats
- Step 4: Verify all Derived Variables

## Exercise 2

- Use CDISC Builder and perform a Difference Test (DIFFTEST) for AE and CM against CDISC Data.
- Document Differences in [define\\_data\\_standards.rtf](#)
- Perform CDISC Evaluation for AE and CM data
- Update Documentation in [define\\_data\\_standards.rtf](#)

## QUIZ 4

- Which of the following variable attributes are **not** used in the data definition documentation?

- A) Variable Name and Type
- B) Variable Length and Label
- C) Variable Formats and Informat
- D) Origins and Roles
- E) Comments



## QUIZ 5

- Which of the following are **not** common ways of applying data standards?
  - A) Used to document internal data standards
  - B) Used to document CDISC data standards
  - C) Document data attributes as target for data standards
  - D) Verify Against data standards after programming is complete.

## Quiz 6

- What are some of the verification steps that are applied for data integrity of data definition?
  - A) Verify hyperlinks of Define.XML
  - B) Verify the order of the variables with keys listed first
  - C) Verify Decoded variables against SAS formats
  - D) Verify all derived Variables
  - E) All the above

# CODE TESTING TASKS

<b>Code Review</b>	Systematic review of program code pertaining to the derivation according to a predetermined checklist of verification criteria.
<b>Code Testing</b>	Perform testing on SAS programs pertaining to the derivation supplying valid and invalid inputs and verify expected output.
<b>Log Evaluation</b>	Evaluate the SAS log for error, warning and other unexpected messages.
<b>Output Review</b>	Visual or programmatic review of report outputs related to the derivation as compared to expected results.
<b>Data Review</b>	Review attributes and contents of output data for accuracy and integrity.
<b>Duplicate Programming</b>	Independent programming to produce the same derivation and output for comparison.



# CODE REVIEW CHECKLIST EXAMPLE

- List of Tasks Ensures Quality

```
SAS PROGRAM
___ Program header complete and up-to-
   date.
___ Program commented clearly and
   adequately.
___ Macro variable definitions and
   usage are correct.
___ Data 'hardcode' corrections clearly
   commented with references in both
   the header and program.
___ Sorting orders correct.
___ Variables have been defined in the
   input datasets.
___ Macros used in more than one
   program are in the tools dir.

SAS LOG
___ Error check program clear.
___ Warning and error messages and
   warnings eliminated.
___ Un-initialized variables and merge
   notes eliminated.
___ Number of observations in each
   dataset as expected.
___ Checked all debugging messages
   (e.g. PUT statements).

SAS OUTPUT
___ All results have be validated
   either by outside methods (e.g.,
   proc freq or means, other) or
   against the CRT file (listing or
   SAS viewer).
___ Titles, footnotes and labels
   verified for accuracy and
   completeness.
___ Spelling is correct. Checked for
   typos.
___ Labels for columns and rows are
   correct.
```

# DOCUMENTATION FORMAT

- Domain Documentation in PDF and XML.
- RTF and XML is Useful for Management



define.pdf



define.rtf



define.xls



define.xml

# DOCUMENTATION STEPS

- STEP 1: Capture all dataset and variable attributes (i.e. PROC CONTENTS) for SAS datasets
- STEP 2: Identification of key fields through sort order of datasets
- STEP 3: Decode all coded lists through a format catalog
- STEP 4: Determine the origins of each variable by identifying matches to variables in source datasets and variables in analysis datasets



# DOCUMENTATION STEPS

- STEP 5: Enter information centrally
- STEP 6: Maintain Audit trail of edits and updates to manage change control.
- STEP 7: Generate DEFINE.PDF and DEFINE.XML from metadata captured
- STEP 8: Update the Information with Update Input Data

## Exercise 3

- Capture Metadata in Excel
- Edit Data Definition in Excel for CM

## COURSE OUTLINE

- XML Primer
- XML Structure by Example - XML CD Example
- Data Definition Documentation Background
- **Define.XML Structure**



# Data Types in DEFINE.XML

ODM Type**	Other Type*	Considerations
text	Char	If the data field were 200 characters (e.g., the SAS Version 5 transport file variable length restriction) then the length would be 200.
integer	Num	Used for numeric or equivalent variables that have discrete whole values (non-fractional) they can be positive, negative, or zero.
float	Num	Use for other non-integer variables where the data type is numeric or equivalent; see the ODM documentation for the significant digit representation.
datetime	Char	Use if values for variable represent complete Date Times (YYYY-MM-DDT HH:MM:SS.SS).
date	Char	Use if values for variable represent complete (YYYY-MM-DD) dates.

# Controlled Terminology (Code List)

<b>Coded Value</b>	<b>def: Rank</b>	<b>Decode</b>
<b>MILD</b>	<b>1.0</b>	<b>Mild</b>
<b>MOD</b>	<b>2.0</b>	<b>Moderate</b>
<b>SEV</b>	<b>3.0</b>	<b>Severe</b>

# Value Level Metadata

Source Variable	Value	Label	Type	Controlled Terms or Format	Origin	Role	Comment
SCTESTCD	ALLERGY	Allergy Status	integer	<u>YESNOUNK</u>	CRF Page		Subject Characteristics CRF Page 4
SCTESTCD	EDLEVLN	EDUCATIONAL LEVEL-DVN	float		CRF Page		Subject Characteristics CRF Page 4
SCTESTCD	EXCLSN	EXERCISE CLASSIFICATION-DVN	float		CRF Page		Subject Characteristics CRF Page 4



# Value Level Metadata Code List

Coded Value	Code Text
YESNOUNK	
Y	YES
N	NO
U	UNKNOWN
NA	NOT APPLICABLE

# EXERCISE

## AUTOMATED TECHNIQUES

- Use Definedoc™ to Generate DEFINE.PDF
- Generate DEFINE.XML
- Edit information and Regenerate DEFINE.XML
- Export to EXCEL and Edit and Update DEFINE.PDF
- Generate a Sample Annotated CRF from Data

# CONCLUSION

- Domain Documentation is Essential for a Review. Cuts Time for FDA Reviewers.
- Also Useful for Programmers and Statisticians: Project Management, Data Integrity and Data Standards
- Constant Data Updates and Documentation Updates Requires Automation
  - Efficiency
  - Accuracy



# QUESTIONS

Sy Truong

MXI Meta-Xceed Inc.

2185 Oakland Rd,

San Jose CA 95131

Phone: (408) 955-9333

Email: [sy.truong@meta-x.com](mailto:sy.truong@meta-x.com)