AGENDA

1. Motivation
2. Automation Overview
3. Architecture
4. Validating the System
5. Pilot Study Results
6. Future State
WHY AUTOMATE?

Return (benefit) = \( \frac{ROI \ (Return \ on \ Investment)}{Investment \ (cost)} \)

- Reduce the per trial cost of submission artifact creation
- Reduce the cycle time for submission artifact creation
- Increase data quality

Source: https://xkcd.com/1319/
1. Insufficient standardization of data sources and targets
   a. Significant progress has been made with the evolution of CDISC and sponsor standards to ensure a consistent approach to data collection and mapping to submission formats

2. Diverse data needs of clinical trials
   a. Science is evolving and new data must be collected to support the science, which results in a constant evolution of standards to support clinical trial data

3. Rigorous regulatory requirements for submission artifacts
   a. ‘Good enough’ is not enough - in many fields machine learning can reduce the effort to map and consolidate data, but the requirements for clinical trial data are more precise and exacting.
SUBMISSION ARTIFACT CREATION
MOVING FROM TRADITIONAL PROGRAMMING TO FULL AUTOMATION

No Automation

Organic Programming Methodology

Traditional programming

- Smallest initial investment
- Most Flexible

Low Efficiency
- Cycle times measured in Months

Why Choose It

Relative Efficiency

Custom Macro Library

- Traditional programming
- Standard macro library for code re-use

Moderate Efficiency
- Cycle times measured in Weeks

Full Automation

Automated Metadata Driven Processing (AMDP)

- Metadata-driven code generating engine

- Repeatable outputs
- Reduced cycle times
- Reduced cost

High Efficiency
- Cycle times measured in Days

- Supports junior level resources
- Enables code re-use
SUBMISSION ARTIFACT CREATION

OUR APPROACH

1. Standardize data collection and target structures
   - As a pre-requisite, we ensure conformance of collected data to CDASH-based collection standards

2. Define precise mapping specifications
   - For each study, we develop machine-readable specifications that define source to target mappings, target metadata, and the sequencing programming steps

3. Develop a macro library
   - Our team created and continues to expand a library of SAS ‘utility’ macros that perform discrete tasks

4. Create a code-generating engine
   - Our team created a framework for generating SAS code based on the mapping specifications and the macro library
# AUTOMATION WITH METADATA

**CDISC METADATA**

- Provides structural metadata to inform dataset structure and variable creation statements
- Ensures compliance with data standards

## Vital Signs Dataset (VS)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Label</th>
<th>Type</th>
<th>Controlled Terms or Format</th>
<th>Origin</th>
<th>Role</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDYID</td>
<td>STUDY IDENTIFIER</td>
<td>text</td>
<td>CRF Page</td>
<td>Identifier</td>
<td>Demographics CRF Page 4</td>
<td></td>
</tr>
<tr>
<td>DOMAIN</td>
<td>DOMAIN ABBREVIATION</td>
<td>text</td>
<td>CRF Page</td>
<td>Identifier</td>
<td>DOMAIN ABBREVIATION</td>
<td></td>
</tr>
<tr>
<td>USUBJID</td>
<td>UNIQUE SUBJECT IDENTIFIER</td>
<td>text</td>
<td>CRF Page</td>
<td>Identifier</td>
<td>Demographics CRF Page 4</td>
<td></td>
</tr>
<tr>
<td>VSSEQ</td>
<td>SEQUENCE NUMBER</td>
<td>integer</td>
<td>CRF Page</td>
<td>Identifier</td>
<td>SEQUENCE NUMBER</td>
<td></td>
</tr>
<tr>
<td>VSTESTCD</td>
<td>VITAL SIGNS TEST SHORT NAME</td>
<td>text</td>
<td>CRF Page</td>
<td>Topic</td>
<td>Vital Signs CRF Page 4, CRF Page 7</td>
<td></td>
</tr>
<tr>
<td>VTEST</td>
<td>VITAL SIGNS TEST NAME</td>
<td>text</td>
<td>CRF Page</td>
<td>Synonym Qualifier</td>
<td>VITALS SIGNS TEST NAME</td>
<td></td>
</tr>
<tr>
<td>VSPOS</td>
<td>VITAL SIGNS POSITION OF SUBJECT</td>
<td>text</td>
<td>CRF Page</td>
<td>Record Qualifier</td>
<td>POSITION OF THE SUBJECT DURING A MEASUREMENT OR EXAMINATION.</td>
<td></td>
</tr>
</tbody>
</table>
AUTOMATION WITH METADATA

‘OPERATIONAL’ METADATA

- Provides instructions for join / merge operations
- Provides instructions for macro calls including parameters values
- Provides instructions on the algorithms used to transform, derive, or compute data.

<table>
<thead>
<tr>
<th>SOURCE_DATASET_NAME</th>
<th>SOURCE_VAR_NAME</th>
<th>RULE_TYPE</th>
<th>DERIVATION_TYPE</th>
<th>DERIVATION_ORDER</th>
<th>RULE_DEFINITION</th>
<th>TARGET_DATASET_NAME</th>
<th>TARGET_VAR_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSPULSE_STD VSPULSE_RAW</td>
<td>S</td>
<td>SC</td>
<td>75</td>
<td></td>
<td>VS_PULSE VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSSSTRESN</td>
<td>S</td>
<td>SC</td>
<td>76</td>
<td></td>
<td>VS_PULSE VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSPULSE_STD UN</td>
<td>D</td>
<td>AS</td>
<td>77 'beats/min'</td>
<td>VS_PULSE</td>
<td>VSSSTRESU</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSPSOS_PULSE_STD</td>
<td>S</td>
<td></td>
<td>78</td>
<td></td>
<td>VS_PULSE VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSMETHOD_PULSE_STD</td>
<td>S</td>
<td></td>
<td>79</td>
<td></td>
<td>VS_PULSE VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSMETHOD_PULSE_STD</td>
<td>S</td>
<td></td>
<td>79</td>
<td></td>
<td>VS_PULSE VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2 WHERE=(VSHR, VSHR_RAW)</td>
<td>D AS</td>
<td>80 'HR'</td>
<td>VS_HR VSTESTCD</td>
<td></td>
<td></td>
<td>VS_PULSE VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSHR_RAW</td>
<td>D</td>
<td>AS</td>
<td>81 'Heart Rate'</td>
<td>VS_HR VSTEST</td>
<td>VS_HR VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSHR_RAW</td>
<td>S</td>
<td></td>
<td>82</td>
<td></td>
<td>VS_HR VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSHR UN</td>
<td>D</td>
<td>AS</td>
<td>83 'beats/min'</td>
<td>VS_HR VSSSTRESN</td>
<td>VS_HR VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSHR_STD VSHR_RAW</td>
<td>S</td>
<td>SC</td>
<td>84</td>
<td></td>
<td>VS_HR VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>RSSSTRESN</td>
<td>S</td>
<td></td>
<td>85</td>
<td></td>
<td>VS_HR VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSSSTRESN</td>
<td>S</td>
<td>DAS</td>
<td>86 'beats/min'</td>
<td>VS_HR VSSSTRESN</td>
<td>VS_HR VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSPSOS HR_STD</td>
<td>S</td>
<td></td>
<td>87</td>
<td></td>
<td>VS_HR VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSMETHOD HR_STD</td>
<td>S</td>
<td></td>
<td>88</td>
<td></td>
<td>VS_HR VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2 WHERE=(VSSBS, VSSBSA_RAW)</td>
<td>D AS</td>
<td>89 'BSA'</td>
<td>VS_BSA VSTESTCD</td>
<td></td>
<td></td>
<td>VS_HR VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSSBSA_RAW</td>
<td>D</td>
<td>AS</td>
<td>90 'Body Surface Area'</td>
<td>VS_BSA VSTEST</td>
<td>VS_BSA VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSSBSA_RAW</td>
<td>S</td>
<td></td>
<td>91</td>
<td></td>
<td>VS_BSA VSSSTRESN</td>
<td></td>
</tr>
<tr>
<td>VS_INPUT_TMP2</td>
<td>VSSBSA RAW</td>
<td>S</td>
<td></td>
<td>92</td>
<td></td>
<td>VS_BSA VSSSTRESN</td>
<td></td>
</tr>
</tbody>
</table>
AMDP ARCHITECTURE

Version 1.0

METADATA SOURCES

- Standards
- Mapping Specs
- Controlled Terminology

METADATA SPECIFICATION

- CDISC Metadata
- Operational Metadata

SCE ENVIRONMENT

AMDP SAS Engine
- Generates

AMDP SAS Library
- Reads
- Instructs
- Generates

Executable Programs
- Executes

Submission-Ready Artifacts
- SDTM
- ADaM
- TFLs

SAS SERVER

Utility Macros
- Reads
- Instructs

www.accenture.com/alsc
1. Created a set of standard artifacts that included representation from the following categories across SDTM, ADaM, and TFLs

**SDTM**
- General Observation Classes
  - Findings
  - Interventions
  - Events
- Special Purpose
- Special Purpose Relationship

**ADaM**
- ADaM Subject-Level Analysis Dataset (ADSL)
- Occurrence Data (OCCDS)
- Basic Data Structure (BDS)

**TFL**
- Basic Listings
- Summary Tables
- Change from Baseline Tables
- Shift Tables

2. Validated the generation of these standard artifacts against data from two different studies
The team received a proprietary global metadata library and source data for the study.

The team then developed study specifications (both human and machine-readable) to define the conversion of the source data into 48 SDTM domains.

Successfully completed SDTM conversion with high quality in 31 business days.

### PILOT STUDY RESULTS

#### Description

- The team received a proprietary global metadata library and source data for the study.
- The team then developed study specifications (both human and machine-readable) to define the conversion of the source data into 48 SDTM domains.
- Successfully completed SDTM conversion with high quality in 31 business days.

#### Quality Metrics

<table>
<thead>
<tr>
<th>Error Category</th>
<th>Error Count</th>
<th>Total Variables</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major</td>
<td>3</td>
<td>700</td>
<td>99.6%</td>
</tr>
<tr>
<td>Minor</td>
<td>6</td>
<td>700</td>
<td>99.1%</td>
</tr>
<tr>
<td>Total</td>
<td>9</td>
<td>700</td>
<td>98.7%</td>
</tr>
</tbody>
</table>

Quality = 1 - \( \frac{\text{Error Count}}{\text{Total Variables}} \)
## PILOT STUDY RESULTS

### Work Breakdown

<table>
<thead>
<tr>
<th>Work Type</th>
<th>Reduction Expected?</th>
<th>Relative Effort (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Specification Interpretation</td>
<td>Yes</td>
<td>10%</td>
</tr>
<tr>
<td>Client-specific Team Training</td>
<td>Yes</td>
<td>5%</td>
</tr>
<tr>
<td>Human-readable Specification Development</td>
<td>Yes</td>
<td>15%</td>
</tr>
<tr>
<td>Machine-readable Specification Development</td>
<td>Yes</td>
<td>40%</td>
</tr>
<tr>
<td>AMDP Code Development</td>
<td>Yes</td>
<td>5%</td>
</tr>
<tr>
<td>QC and Compliance Reviews</td>
<td>No</td>
<td>15%</td>
</tr>
<tr>
<td>Client and Project Management</td>
<td>No</td>
<td>10%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

### Takeaways

- Time to perform most tasks expected to decrease with study volume
- Very little Code Development
- 55% of effort devoted to specification development
<table>
<thead>
<tr>
<th>Step</th>
<th>Task Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specification Development: Streamline the user experience for creating metadata specifications by creating an intuitive user interface</td>
</tr>
<tr>
<td>2</td>
<td>Metadata Model: Simplify the metadata needed to support generation of outputs, while retaining needed functionality</td>
</tr>
<tr>
<td>3</td>
<td>Code Generating Engine: Re-write the engine in a general-purpose language such as JAVA to support the scalability and maintainability of the system</td>
</tr>
</tbody>
</table>
AMDP ARCHITECTURE

Current State

METADATA SOURCES

- Standards
- Mapping Specs
- Controlled Terminology

User Interface

METADATA SPECIFICATION

- CDISC Metadata
- Operational Metadata

In-Development

Updating and enhancing

SCE ENVIRONMENT

AMDP SAS Engine

Generates

Executable Programs

Executes

Submission-Ready Artifacts

AMDP SAS Library

Utility Macros

SDTM  ADaM  TFLs

Instructs

Reads

www.accenture.com/alsc
Questions?

Silas McKee
silas.a.mckee@accenture.com