PhUSE 2016

Paper DH08

# Efficiency Comes From Reusability and Repeatability

Hanming Tu, Accenture, Berwyn, USA
Dave Evans, Accenture, Berwyn, USA

## ABSTRACT

Building reusable mapping between raw and standard data sets and repeatable process in data handling is important in improving the efficiency of data integration and standardization. Reusability and repeatability are two important measurements of efficiency in the extraction, transformation and loading (ETL) process. This paper discusses the process of extracting and transforming raw data into CDISC SDTM standard and demonstrate how to create reusable maps and repeatable workflows using the AutoDCD system – an automatic data conversion development system to convert raw data into SDTM data sets.

## INTRODUCTION

CDISC SDTM Data Conversion is part of an ETL (extraction, transformation and loading) process. It requires to develop new ETL packages for each study and produces the same data structure and maintain correct relationships for all the studies. How could we generate ETL packages that produce the quality SDTM data sets from vastly heterogeneous sources with high efficiency?

It usually involves in understanding the protocol, analyzing source data structure, mapping the source with SDTM data model, building transformation packages, and verifying the final SDTM data sets. The SDTM data conversion is not a typical ETL project. In a typical ETL project, you analyze, build, deploy and support your conversion packages. The packages are maintainable codes. In a SDTM data conversion project, your final products are the deliverable SDTM data sets. We will only focus on creating reusable mapping and defining repeatable workflows.

## CREATE MAPPING

The first thing that we need to do is to map source to target. Mapping the source to target requires the good understanding of the target – the CDISC SDTM, the source – the data sets that you got from your clients and the relationships among them. In general there are two levels of mapping: data set (domain) and variable (column).

### MAP DATA SET TO DOMAIN

There are four types of SDTM Mapping at domain level:

- `Simple match`: one to one
- `Converging match`: many to one
- `Diverging match`: one to many
- `Topic based match`: domain aggregation

The one-to-one matching matches single dataset with single SDTM domain. It is simple but very rare, and only if clinical data management system (CDMS) is SDTM compliant but still not 100%. The example domains are Trial Arms (TA), Trial Elements (TE), Trial Visits (TV), Trial Inclusion/Exclusion Criteria (TI), and Trial Summary (TS).

The many-to-one matching is a converging mapping - records from many different datasets merged into one SDTM domain. It is complex and common and usually represents a dependent relationship. The example SDTM domains or classes are

- RELREC: AE, CM, LB, MB, MS through studyid, rdomain, usubjid, idvar, idvarval, reltype, relid columns
- SUPPQUAL: SuppAE, SuppCM, SuppDM, SuppEG, SuppEX, SuppMH, SuppPE, etc.

The one-to-many matching is diverging mapping - records from one dataset split into many SDTM domains. It is common and complex. The example SDTM domains are

- Demographics – DM → DM, DS, SC, SUPPDM
- Laboratory Test Results – LB → LB, CO, SUPPLB

The topic based match is a type of domain aggregation. The datasets with topic variables will be consolidated into SDTM observation classes, for instance:

- The TRT topic domains to Interventions
- The TERM topic domains to Events
- The TESTCD topic domains to Findings

This has to be done for every study; then all the studies are pooled into one in the multi-study project.

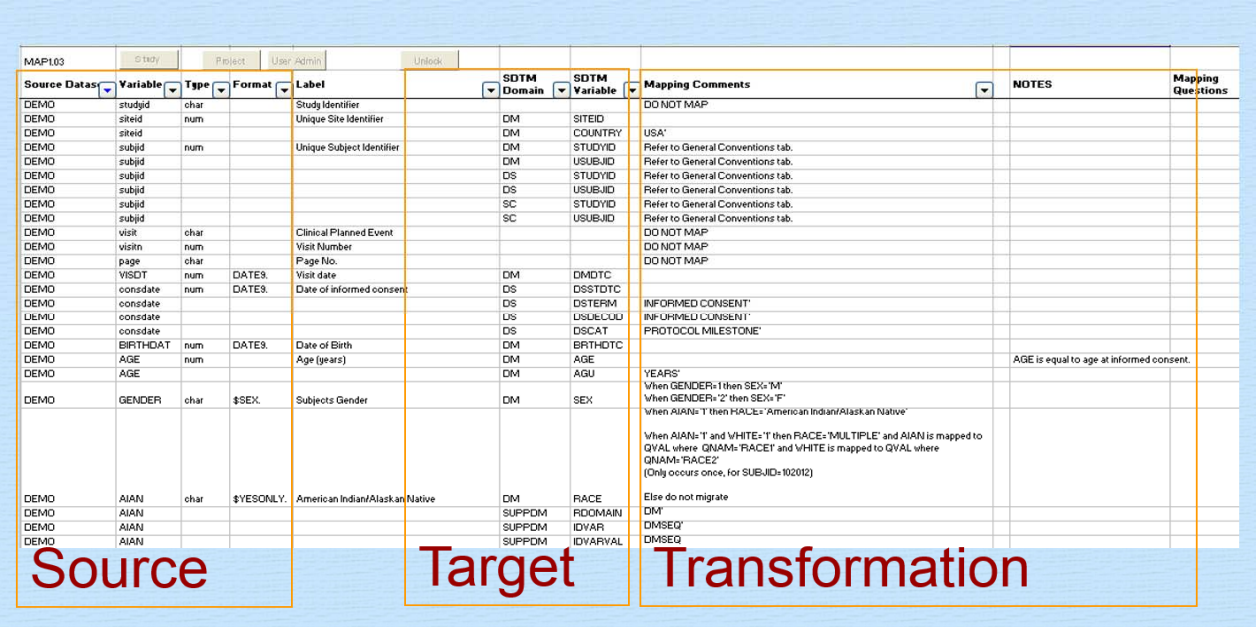
**MAP VARIABLES TO COLUMNS**

At the variable level, we need to map and transform the data and metadata.

- Metadata transformation
  - `Name`: matched or renamed
  - `Data type`: matched or casted
  - `Length`: contained or split
  - `Label`: matched or changed
- Data transformation
  - `Decoding` or `encoding`: translating, controlled terms, lookup – value to code or vice versa
  - `Combining` or `splitting`: many fields to a column or vice versa
  - `Transposing` or `pivoting`: rows to columns or vice versa
  - `Selecting` or `filtering`: variables or records
  - `Aggregating` or `deriving`: new variables
  - `Generating`: surrogate-keys

**MAP SPECIFICATION EXAMPLE**

The following figure shows an example of mapping the source to target and the transformation needed to convert the data into the target.




| Source Datas | Variable | Type | Format | Label | SDTM Domain | SDTM Variable | Mapping Comments | NOTES | Mapping Questions |
|---|---|---|---|---|---|---|---|---|---|
| DEMO | studyid | char | | Study Identifier | | | DO NOT MAP | | |
| DEMO | siteid | num | | Unique Site Identifier | DM | SITEID | | | |
| DEMO | siteid | | | | DM | COUNTRY | 'USA' | | |
| DEMO | subjid | num | | Unique Subject Identifier | DM | STUDYID | Refer to General Conventions tab. | | |
| DEMO | subjid | | | | DM | USUBJID | Refer to General Conventions tab. | | |
| DEMO | subjid | | | | DS | STUDYID | Refer to General Conventions tab. | | |
| DEMO | subjid | | | | DS | USUBJID | Refer to General Conventions tab. | | |
| DEMO | subjid | | | | SC | STUDYID | Refer to General Conventions tab. | | |
| DEMO | subjid | | | | SC | USUBJID | Refer to General Conventions tab. | | |
| DEMO | visit | char | | Clinical Planned Event | | | DO NOT MAP | | |
| DEMO | visitn | num | | Visit Number | | | DO NOT MAP | | |
| DEMO | page | char | | Page No. | | | DO NOT MAP | | |
| DEMO | VISDT | num | DATE9. | Visit date | DM | DMDTC | | | |
| DEMO | consdate | num | DATE9. | Date of informed consent | DS | DSSTDTC | | | |
| DEMO | consdate | | | | DS | DSTERM | 'INFORMED CONSENT' | | |
| DEMO | consdate | | | | DS | DSDECOD | 'INFORMED CONSENT' | | |
| DEMO | consdate | | | | DS | DSCAT | 'PROTOCOL MILESTONE' | | |
| DEMO | BIRTHDAT | num | DATE9. | Date of Birth | DM | BRTHDTC | | | |
| DEMO | AGE | num | | Age (years) | DM | AGE | | AGE is equal to age at informed consent. | |
| DEMO | AGE | | | | DM | AGU | 'YEARS' | | |
| DEMO | GENDER | char | $SEX. | Subjects Gender | DM | SEX | When GENDER='1' then SEX='M'<br>When GENDER='2' then SEX='F' | | |
| DEMO | AIAN | char | $YESONLY. | American Indian/Alaskan Native | DM | RACE | When AIAN='1' then RACE='American Indian/Alaskan Native'<br><br>When AIAN='1' and WHITE='1' then RACE='MULTIPLE' and AIAN is mapped to QVAL where QNAM='RACE1' and WHITE is mapped to QVAL where QNAM='RACE2'<br>(Only occurs once, for SUBJID=102012)<br><br>Else do not migrate | | |
| DEMO | AIAN | | | | SUPPDM | RDOMAIN | 'DM' | | |
| DEMO | AIAN | | | | SUPPDM | IDVAR | 'DMSEQ' | | |
| DEMO | AIAN | | | | SUPPDM | IDVARVAL | DMSEQ | | |

Figure 1: Map Specification Example

## BUILD REUSABLE CODES

There are different ways to build reusable codes. It could be opportunistic or planned. An opportunistic approach is ad hoc way of finding or creating reusable codes. A planned approach brings systematic code reuse. The codes developed for the datasets in one phase of a clinical study could be used for all the phases or from one study to be used for all the studies in the same therapeutic area. It is a strategy for increasing productivity and improving quality in the data transformation and standardization. Although it is simple in concept, successful code reuse implementation is difficult in practice.

Traditionally code reuse is done through leveraging the code libraries or adding custom codes into the libraries.

Oracle Warehouse Builder (OWB) is a single, comprehensive tool for data integration and it provides data quality, data auditing, fully integrated relational and dimensional modeling, and full lifecycle management of data and metadata. The issue with using OWB is the productivity. The design center in OWB is visual but take a lot of time and requires big screens to allow users to draw lines to link the sources and targets after we have built mapping specifications. After using OWB for three years, we built a much faster web-based system to automate the data conversion development (DCD). Here is the comparison among the traditional approach, OWB approach and AutoDCD approach:

| Traditional Approach | OWB Approach | AutoDCD |
|---|---|---|
| ETL using custom programming such as SAS, PL/SQL, JAVA, Perl, etc. | ETL with User Interface | Web-based User Interface |
| High paid programmers | Users do not need to know the programming language – PL/SQL | No PL/SQL programming is needed |
| No audit trail | In an audited environment | In an audited environment and AutoDCD is validated product |
| No security | Built-in security: database and OWB security | Authenticated and authorized users only with audit trails |
| No consistence among coding | Consistence with all the users | Automatic and consistent coding |
| Difficult to manage and support | Easy to manage and support | Easy to manage and support |
| Scalability: silo and not scale; through adding more manpower | Scalability through hardware and software | Very scalable |

Figure 2: Comparison among ETL Tools

Here are a few important considerations for increasing the code reusability:

- Standard adoption is the key for code reusability
  - Train people to understand the standards
  - Define standard templates
  - Build public libraries for code snippets and public transformation: Custom functions, procedures and packages; public data rules; and public Experts
  - Group code snippets and functional transformation into modular mapping and transformation: pluggable maps
  - Define workflow to govern the process: Workflow Manager and Process Flows
- Metadata-driven process is the key for automation
  - Metadata makes data meaningful
  - Metadata is machine readable
  - Metadata is the base for automation
- Replication and automation are the focuses
  - Use or create utilities to replicate the process: OMB+ for Project Set Up, Mapping Specification, Mapping Creation
  - Use analytics tool to identify the areas for replication and automation: Data Profiling & Data Rules for Source Data Review / Edit Checks

When we have identified the components that can be reused, we can follow the following steps to build reusable modules or even a base project which can be used to start a new project.

- Extract common components:
  - Build a public code library
    - Transformation
    - Utilities: functions, procedures, packages, pluggable maps, workflows
  - Build metadata repository:
    - SDTM data model
    - Controlled terminologies
    - Specification lookup tables: mapping intelligence
  - Create a base project
    - Common modules

- Public locations (database links)

- Build subsequent projects
  - Create location linking to metadata repository
  - Import public utilities: transformation, data rules and experts
  - Copy the base project and modules

The key of the reusability and automation is the metadata. Here is the metadata model used in AutoDCD:

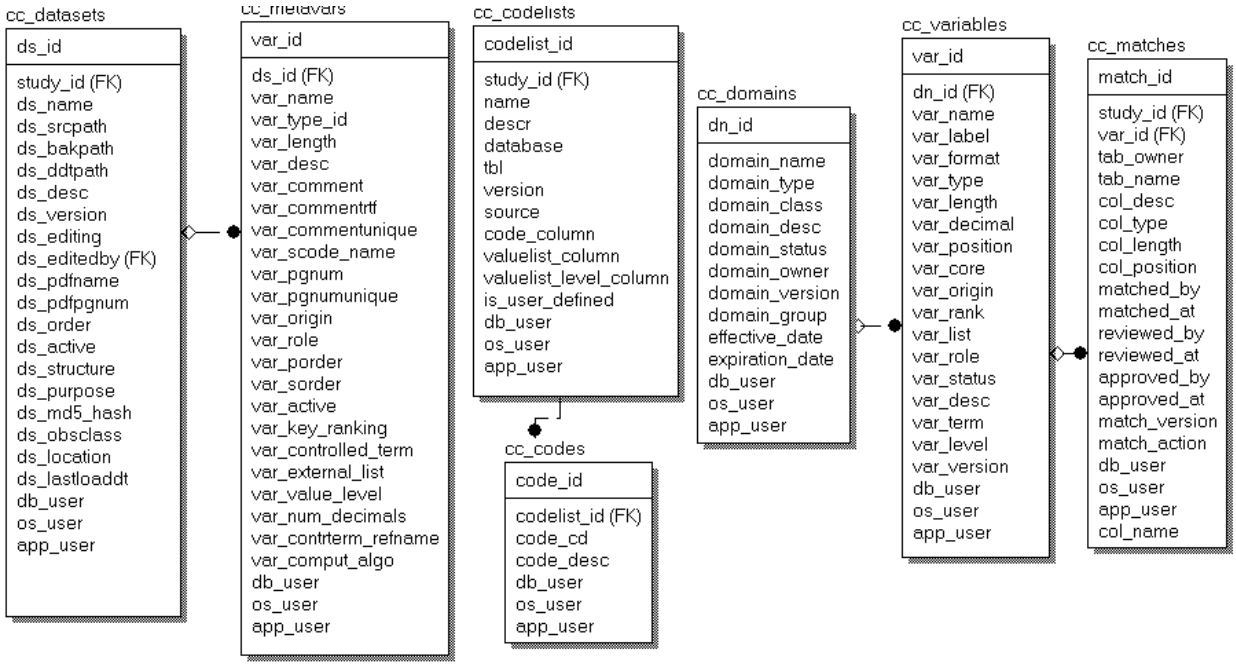



Figure 3: Metadata Repository Data Model

## DEFINE WORKFLOWS

We could gain efficiency by building repeatable workflows. A defined and repeatable workflow also ensure the quality of the work. The following diagram shows the overall process of data conversion. In reality, we have so many isolated systems and single purposed codes. How could we link reusable silo codes into a repeatable process?

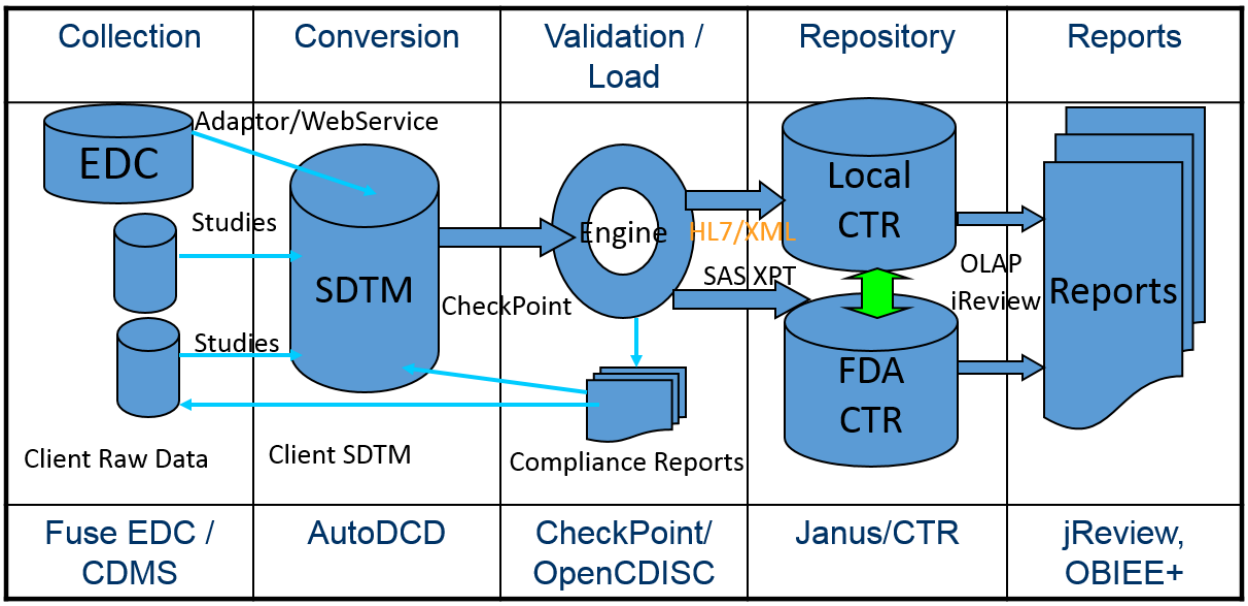



Figure 4: Conversion Overall Process

We built an automatic data conversion system to link the pieces in data integration and standardization process. AutoDCD provides the workflow to link maps to form a controlled data flows, from vertical code reusability to horizontal process repeatability. The AutoDCD portal provides a web-based software tool for the Data Integration and Standardization (DIS) Department to use. The portal improves and accelerates the Data Conversion Development (DCD) process.

The input to AutoDCD is a Data Conversion Specification (DCS) that describes the source to target migration process. AutoDCD extracts and loads into Oracle the SQL statements from the DCS and sequences these statements into a "stored procedure". The 'stored procedure' is then executed to convert source data into domains compliant with the CDISC SDTM model.

AutoDCD is a three tier web-based application: Oracle database, Oracle application server (Apache web server) and a browser. Here is a list of features being implemented:

- Load and store map specification in relational database
- Manage workspace with client, project, study, and specification hierarchy
- Allow users to create and delete intermittent views and tables that we used in mapping
- Run data conversion jobs by domain or by a group of domains or all
- Link and copy tables from an Oracle database or use the tables created and loaded through SAS upload utility
- Keep audit trails for each job
- Track the performance of each job

In AutoDCD 3.0 design, a SAS service will be added to integrate with SAS scripts including upload (import) and download (export) SAS macros. See the following data flow diagram.

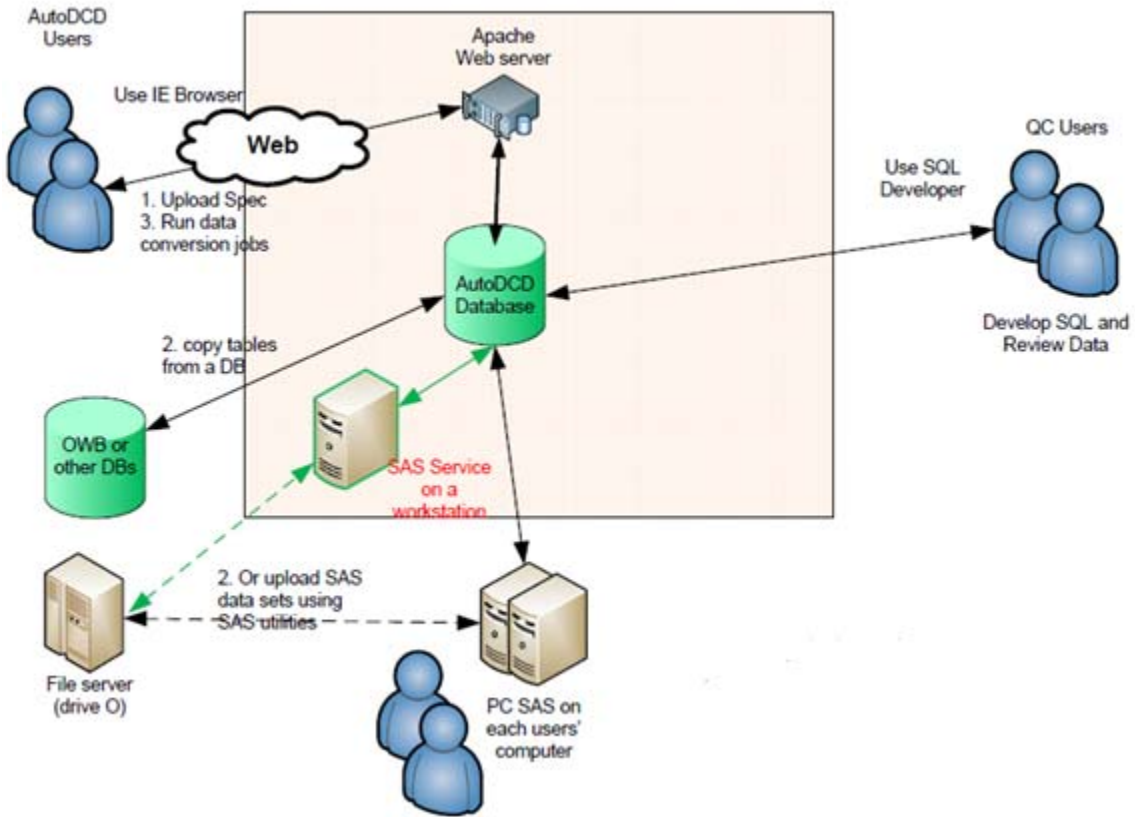



Figure 5: AutoDCD Data Flow Diagram

The important value of the AutoDCD is the automation of the data transformation process. Here is a list of components that can be modularized and be replicated by utility programs or tools.

| | Specification | Development | Deployment | Execution | Validation |
|---|---|---|---|---|---|
| **Method** | Specification template / match lookup table | Base project, Pluggable maps, maps, packages, and workflows | Generating PL/SQL packages | Workflow module, template | SDTM compliance Checks (FDA) |
| **Products** | Map specification tool, AutoDCD spec loader | Map authoring tool, OMB+, OWB expert, AutoDCD | OWB, OMB+, PL/SQL package, AutoDCD | Workflow manager, OWB expert, DB Jobs, AutoDCD Jobs | CheckPoint, OpenCDISC validator |
| **Technologies** | **Oracle Appex** | **OMB+, OWB Expert Tool** | **OWB** | **OWB, Workflow manager** | **Oracle + Validation engine** |

Figure 6: Reusability Matrix

## CHECK THE COMPLIANCE

Once we produced the target data sets, how could we verify whether the data sets comply to the standards or the defined quality? How do we measure the quality of the work? We need to have some sort of governance in place to ensure the quality of data sets and set of metadata to describe the quality of the data. Clinical data quality information is semantic information about clinical data quality (CDQ), including how a clinical trial is conducted and how the data elements are collected, entered, processed, and analyzed. CDQ concerns not just data accuracy, but also data traceability and compliance against common standards. The industry often uses six sigma concept. It is out of the scope of this paper to discuss the six sigma in clinical data management. We rather just focus on defining a CDQ measurement and discussing how we could use a set of compliance checks to ensure the target data conforming to CDISC SDTM standard. There are generally two approaches for CDQ: system-based and standard-based approaches.

In a system-based approach, CDQ can be assessed based on the error or failure rates.

- Paper-based (DDE): In a double data entry system, the data quality can be reflected by the error rate as shown in these two examples:
  - Source-to-database error rates: 976 errors per 10,000 fields including medical record abstraction
  - CRF-to-database: 14 errors per 10,000 fields
- Fax-Based (OCR): In a fax based system, the data will be collected initially through optical character recognition capability of the system, then checked by data coordinators. The CDQ could be measured by re-fax rate.
  - Re-fax Rate: 5~20%
- EDC systems: In an electronic capture system, most the error could be captured by system edit checks and then corrected by users.
  - Source-to-database: 50 errors per 10,000 fields

In standard-based approach, we can have the CDQ based on the process, data and verification standards:

- Process standards: GxP where x=L, C, M, etc.
- Data standards:
  - Structure: CDISC ODM, CDASH
  - Value (metadata and controlled terminology):
    - ISO 11179 – IT -- Metadata registries (MDR)
    - ISO 20943 - IT -- Procedures for Achieving Metadata Registry Content Consistency
    - ISO 21090 - Healthcare Data types
    - CDISC SHARE
  - Content: CDISC SDTM
- Verification standards:
  - Validation checks: 99 compliance checks

- JANUS checks: FDA checks for data loading
- Severity Definition: Low, Medium and High

It is not easy to define CDQ measurement when there is no common standard. Many practitioners and researchers have tried to quantify clinical data quality by studying the process of collecting data elements in a clinical trial. More proper approach should be based on standards, treating the process as a whole and defining a composite index. CDQ index is a composite measurement for the clinical data and the process of collecting them. We can define the CDQ Index as *CDQI = PQI + DQI,* where PQI is process quality and DQI is data quality index It might not be easy to define a quantitative measurement for CDQI but many attempts have been made to use other indicators to quantify data quality. According to six sigma, we could define the CDQI as defects per million opportunities (DPMO), and here is the formula for it:

$$DPMO = 1000000 * \sum_{t=1}^{n} \sum_{i=1}^{m} \frac{e_i}{d_i}$$

Where i is a domain, a class or a study in SDTM; m is total number of domain, classes or studies. The t is an investigator site, a project, a client or a compliance check category; n is total number of sites, projects, clients or compliance categories.

$$e_i = \sum_{c=1}^{C} error_c$$

Where c is each compliance check; C is total compliance checks being run for a domain, a class, or a study in SDTM. Error is number of issues being found.

$$d_i = \sum_{j=1}^{J} C_j R_j$$

Where j is an observation or row of data; J is total number of observations or records being run through a compliance check. C is number of columns or variables; R is number of rows or observations.

Figure 7: CDQ Index Calculation

How could we find out the defects in clinical data? We can run through the converted data sets through a set of compliance checks. You could use OpenCDISC validator or some other tools. We have built a system called CheckPoint to conduct such checks. Here are the categories of checks in the system:

- **Consistency**: Checks data in 2 or more columns to ensure data correspond in cross-column (visit number without visit description, age unit without age), cross-domain (SUBJID in a domain but not in DM) or cross-system (external dictionary).
- **Format**: Checks if data are in an allowable format such as ISO 8601; Leading and trailing spaces; and missing value "." in character column
- **Limit**: Checks if data are within range such as start/end time, and toxicity grade.
- **Metadata**: Checks if tables and columns have valid metadata
- **Presence**: Checks data that are missing or present
- **Referential**: Checks if a described table/record data relation are valid; SDTM is not 3NF, referential information is stored as data (RDOMAIN, USUBJID, IDVAR, IDVARVAL) , in Supplemental Qualifiers (SUPPQUAL), related records (RELREC) and comments(CO).
- **Value**: Checks data against valid values such as code lists, Illegal values

Here is a list of the checks in each category:

| Categories | Standard FDA | Enhanced |
|---|---|---|
| Consistency | 46 | 38 |
| Format | 6 | 53 |
| Limit | 10 | 2 |
| Metadata | 8 | 45 |
| Presence | 4 | 33 |
| Referential | 8 | 4 |

| | | |
|---|---|---|
| Value | 27 | 42 |
| **Σ** | **109** | **217** |

Figure 8: CDQ Index Categories

Composite data quality index provides a comprehensive measurement for the quality of clinical data. It will have to start with adopting common standard. Sticking to a data quality model will ensure to improve data quality and the business bottom-line. Criteria based on SDTM standard provides a good measurement for clinical compliance and submission readiness. It is important to insert standard-based verification earlier in the clinical data collection and analysis.

## CONCLUSION

CDISC SDTM is stable and recommended data model for interchange between sponsors and FDA. Standard based approach enables code reusability and process repeatability to gain greater efficiency and consistence. ETL tool like AutoDCD provides security, scalability and audited environment. High data quality relies on standard process, mature technologies, and trained people. So as it is shown in the efficiency level matrix, standard-based systems allow for integration while metadata-driven systems enable automation. The more intelligence collected about the clinical data, the more integration could be; the more integration, the more metadata-driven automation is required; the more standards are adopted, the more meaningful metadata could be applied in the process and the more efficient the process could become.

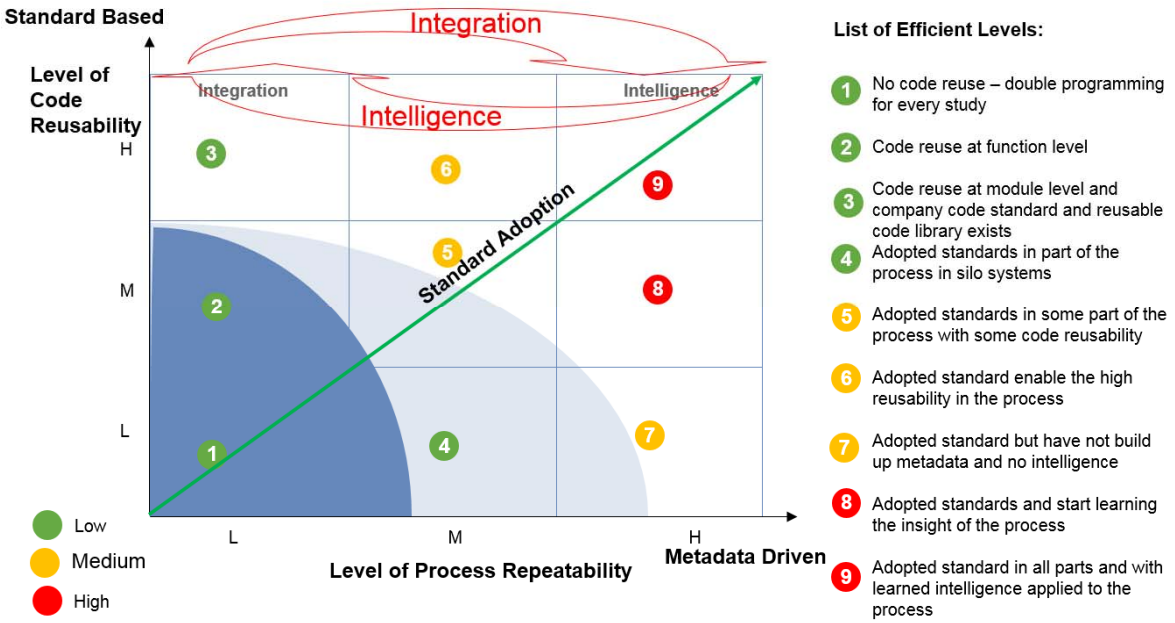



Figure 9. Efficiency Level Matrix

Further automation could be achieved through “Intelligent Data Flow”. The code reusability and process repeatability makes the data conversion very fast; the compliance check ensures the quality is good; the intelligent data flow makes the whole clinical data lifecycle smart! So it makes the overall project relatively cheap.

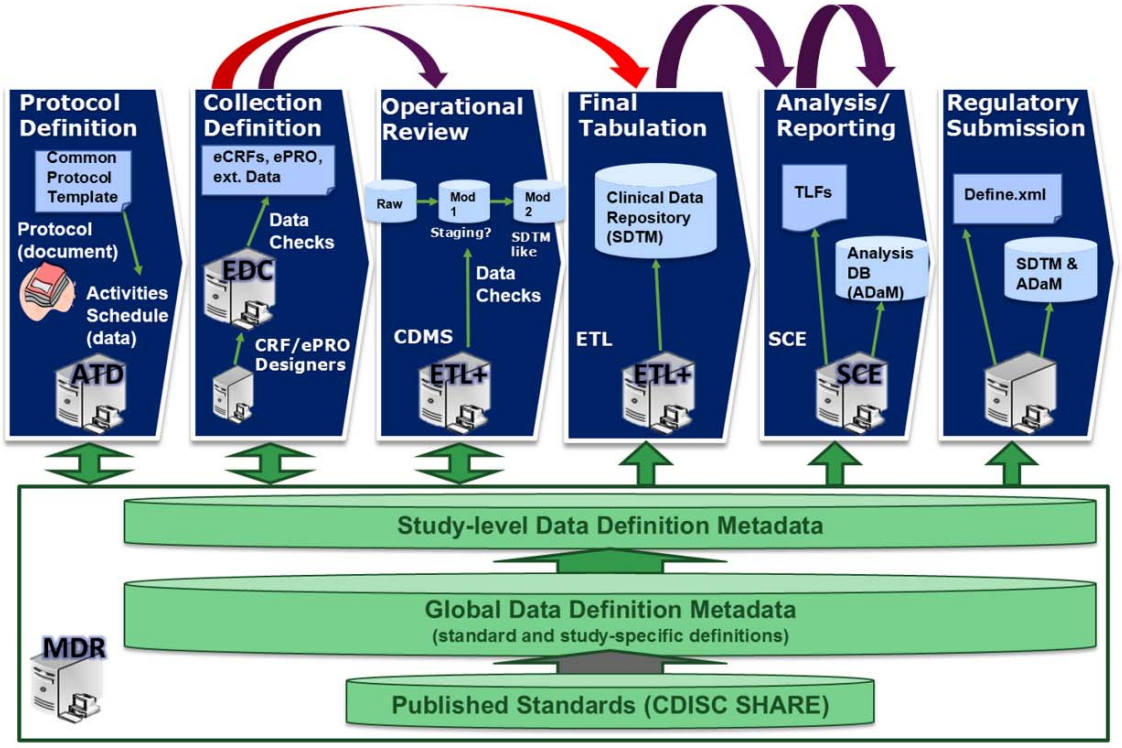



Figure 10. Further automation through “Intelligent Data Flow”

## REFERENCE


1. Bretherton, F. P.; Singley, P.T. (1994). "Metadata: A User's View, Proceedings of the International Conference on Very Large Data Bases (VLDB)". pp. 1091–1094.
2. Kimball et al., *The Data Warehouse Lifecycle Toolkit*, Second Edition. New York, Wiley, 2008, ISBN 978-0-470-14977-5, 116–117
3. NISO. *Understanding Metadata*. NISO Press. ISBN 1-880124-62-9. Retrieved 5 January 2010.
4. ISO/IEC 11179-1:2004 Information technology - Metadata registries (MDR) - Part 1: Framework
5. White paper from Octagon Research Solutions, Inc., “Metadata Management in Clinical Research”, 2010
6. Hanming Tu, June 1, 2009, “Data Management: Information Integrity”, PharmaAsia online magazine
7. Hanming Tu, December 25, 2008, “Roadmap for Clinical IT at Octagon Research”, an internal white paper
8. Hanming Tu, November 1, 2008, “Getting Up to Standard”, PharmaAsia online magazine

## CONTACT INFORMATION

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


| | |
|---|---|
| Author Name: | Hanming Tu |
| Company: | Accenture |
| Address: | 1160 West Swedesford Road |
| City / Postcode: | Berwyn, PA 19312 |
| Work Phone: | 610-407-1817 |
| Fax: | 610-535-6615 |
| Email: | hanming.h.tu@accenture.com |
| Web: | www.accenture.com |

| | |
|---|---|
| Author Name: | Dave Evans |
| Company: | Accenture |
| Address: | 1160 West Swedesford Road |
| City / Postcode: | Berwyn, PA 19312 |
| Work Phone: | 484-881-2411 |
| Fax: | 610-535-6615 |
| Email: | dave.a.evans@accenture.com |
| Web: | www.accenture.com |


Brand and product names are trademarks of their respective companies.