



CDISC India Day

Designing the Future of Clinical Data Workflows

Transforming CDISC standards into actionable user experiences - practical UX patterns across the lifecycle.



Narendra Parmar
Sycamore Informatics



Speakers



Narendra Parmar
Lead UX Designer
Sycamore Informatics

Narendra Parmar is a **Lead UX Designer** at **Sycamore Informatics** with over a decade of experience designing enterprise software and AI-powered digital products. An IIT Bombay graduate in HCI, he specializes in simplifying complex clinical data workflows through human-centered design. Narendra is passionate about trustworthy AI, enterprise UX, accessibility, and transforming complex systems into intuitive experiences that improve productivity and decision-making.







CDISC is not
difficult but working
with **TOOLS** is

SDTM

ADaM

define.xml

CDISC
Implementation
Guide

CDASH

The Real Problem

The Reality of Designing a Protocol

Imagine you're a protocol author designing a new protocol. Working with CDISC standards makes sense. But here's the reality:

- Jumping between **multiple documents**
- Manually mapping **hundreds of data fields**
- Comparing against standards from **different domains**
- Tracking what applies to **YOUR study** vs. boilerplate
- Honestly? You only deliver on **20% of those fields**

The real friction: Multiple Documents, Spreadsheets, Systems and information embedded in review comments.

Wasted Weeks

The Downstream Impact

When every design decision is manually negotiated and context is repeatedly lost:

- **Days researching** what applies to your specific trial
- **Manual comparisons** across multiple rigid documents
- Constant **context switching** between fragmented systems
- **Study design delays** that compound exponentially downstream

The Real Problem

For Statistical Programmer setting up SDTM mappings, it's a different challenge. When you're creating mappings for your study, the standard mappings don't account for your study variation. You're:

- Referring to **multiple Excel sheets**
- Comparing **different mapping levels** (variable-to-variable, domain, study, form)
- Asking: "**For MY study variant, what exactly do I need?**"
- Finding that answer is **buried across multiple systems**

The real friction: Mapping context is fragmented.

Manual Discovery

When standard templates fail to reflect specific study variations, execution stalls:

- **Weeks comparing** reference materials and specifications
- **Error-prone manual filtering** across disparate sources
- **Study variation** creates redundant design rework
- **Significant delays** to the downstream execution timeline

The Real Problem

For Statistical Programmers executing database locks, there's an added pressure: time. When you run a batch of SAS programs and something breaks after its completed, you've lost time. You're:

- **Manually reviewing** error logs
- **Tracing** where things went wrong
- **Starting from scratch** if bugs are found
- **Racing** against submission deadlines

The real friction: locating the issue is difficult.

The Debugging

When feedback loops are excessively long, timeline execution stalls:

- **Debugging** takes a lot of time
- **No visibility** into which domain broke
- **Manual error log review** is time-consuming
- **Submission timeline** is now at risk

The Law of Conservation of Complexity

*“Every application has an inherent amount of irreducible complexity.
The only question is: Who will have to deal with it, the user or the product team?”*

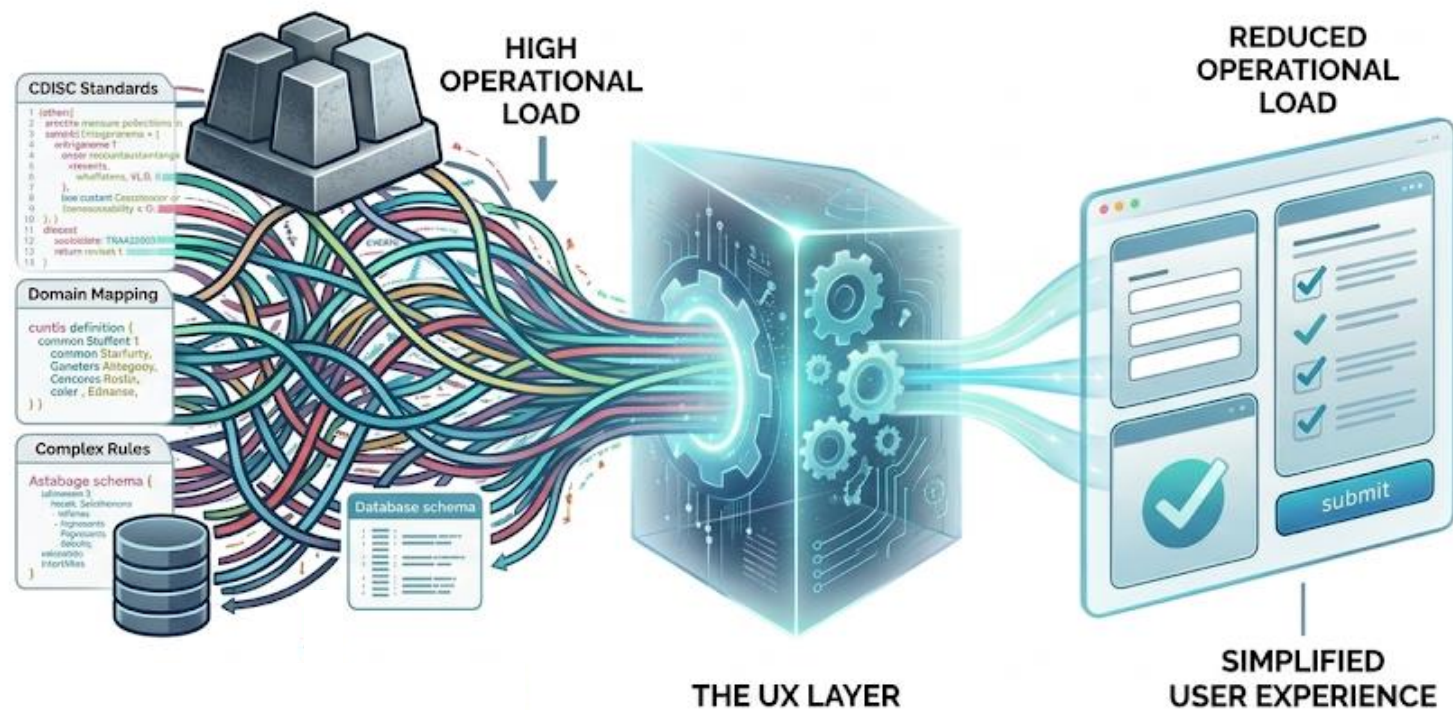


Larry Tesler

Computer Scientist & Pioneer of Interaction Design

Our Philosophy

Shift the complexity from the ~~user~~...
...to the product.



WITHOUT UX

Users must learn the software.

Cognitive load falls on the programmer.



WITH UX

Software understands the user.

Operational load is absorbed by the tool.

Transformation 1: Study Design (SPD)

The Problem

Imagine you're a **Protocol Author** on day one of a new study. You need to set up your protocol for a promising drug for a Phase 1 study in a short time frame.

You're:

- Start with templates from past studies manually
- Find similar protocols, Find and copy what applies to YOUR protocol manually
- Moving between Objectives, Endpoints, Estimands, Validation Rules...
- Losing context constantly

Minutes turn into hours and hours turns into days.

The Solution: Visual Workflow

All the Past Protocols are available in the Structured Protocol Library and Public Registries (clinicaltrials.gov) as a starting point. We show all the relevant studies in context (Phase, Indication, Drug, Objectives, etc) We show the Past and New Protocols USDM Structured document to the user. The system presents pre-populated study configurations (CDISC CT in USDM compliant UI)—to reuse proven patterns.

Instead of switching tabs constantly, one continuous experience maintains context:

 Sticky Navigation

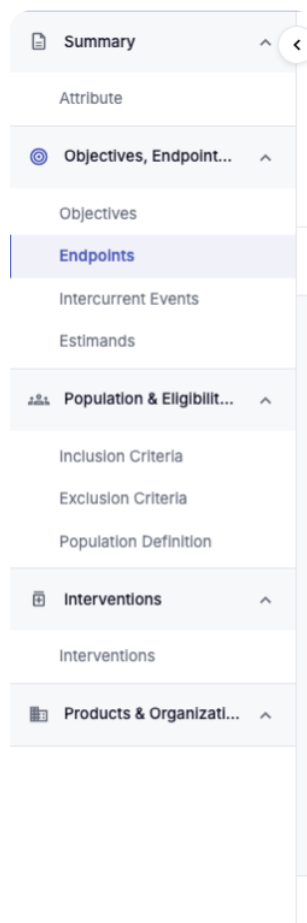
 Progress Indicators

 Change Tracking

 Batch Save

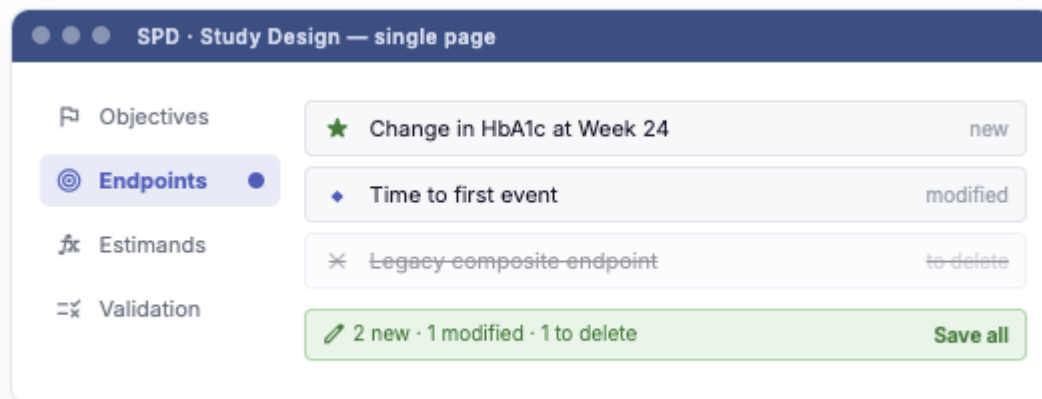
THE RESULT: PROTOCOL SETUP GOES FROM WEEKS TO HOURS.

Right context + smart filtering = faster decisions.

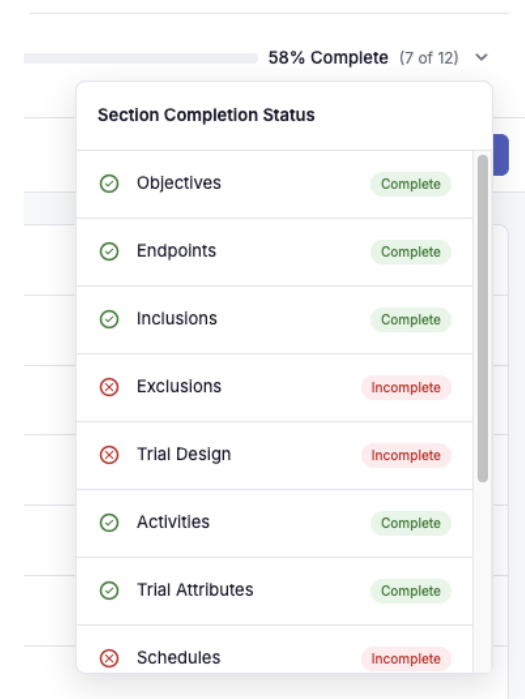


USDM Structured
Sticky Navigation

Progress Indicator



Change Tracking



Transformation 2: Mapping Specifications (MDR)

The Problem

Day One for a programmer.

They are asked to create SDTM mappings.

Traditional tools show:

- 300+ fields
- Blank, flat grids
- No CDISC rules, MSG validations

You either know CDISC, or you guess.

The Solution: **Guided Metadata**

Instead of a blank grid, the intelligent grid with filters based on **YOUR study context**.

The system doesn't ask questions. It shows you:

- ✓ **Standard mappings** relevant to YOUR study
- ✓ **Domain levels** you actually need
- ✓ **Only the variables** for your forms

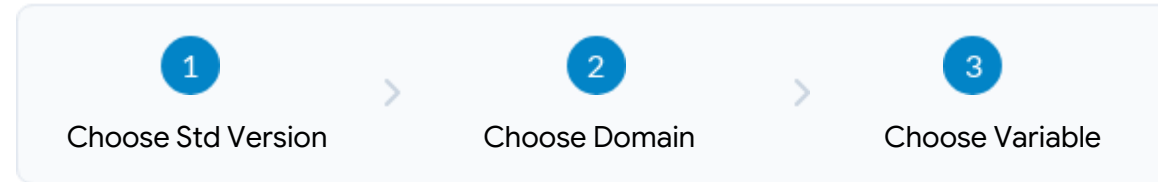
Advanced filtering means: "What does MY study actually need?"

THE RESULT: Weeks to Hours

What would take weeks of comparing reference materials now happens in hours for a first pass draft —because context is built in.

UX doesn't simplify CDISC. It simplifies conforming to CDISC in the context of YOUR study.

Guided Metadata creation



Conformance Wizard

✓ Version ✓ Domain 3 Variables 4 Review

✓ SDTMIG 3.4 · Domain: VS (Vital Signs)

VS TESTCD	CT: VSTESTCD ✓
VSORRES	required
VSORRESU	CT: UNIT ✓

i Rule SD1082 applies to this domain — shown as you go

Transformation 3: DB Lock to Top-Line Results (SCE + DSW)

The Problem

The Database Lock Crunch

For Statistical Programmers, database lock is a high-pressure moment with multiple SAS programs to produce Top-Line and Final Results:

- **One for each data domain** (AE, DM, VS, LB...)
- Each program picks domain-specific data from EDC/labs
- Each outputs a CSV file
- All executed and reviewed under tight deadlines

The breakdown: Errors surface too late

You kick off the batch. **45 minutes later, it fails.** Now you are hunting through logs to find out which domain broke and why while the deadline looms.

The Solution:

Design Philosophy: Execution should be automatic.

Errors should surface upfront.

1. **Dedicated Execution:** Programs run in an environment with exact R/Python versions (DSW container management).
2. **Upfront Error Diagnostics:** Detailed failure info is delivered immediately—not after 45 minutes of waiting.
3. **Visible Lineage:** Trace exactly which dataset produced which output.

The Human Part

You submit and let the system do the heavy lifting. No sitting there monitoring batch jobs.

RESULT: Get to your Top-Line Results faster with auto-correct loops in case of errors.

🕒

Execution Failure — Diagnostic Chain

🕒

Execution Context

JOB ID

job-7f4a2b8c-e91d

USER

analyst@company.com

SUBMITTED

2026-08-10 14:32:17 UTC

R VERSION

4.3.1

CONTAINER

rwe-analysis:v2.1.4

ENVIRONMENT

production

🟢

Program Syntax Check

✓ PASSED

R script syntax validated. No parse errors detected in `analysis.R`.

🟢

Environment & Dependencies

✓ LOADED

Container launched with pinned R 4.3.1 + 12 dependencies (tidyverse 2.0.0, ggplot2 3.4.2, etc.). All packages initialized.

🟢

Input Data Load

✓ COMPLETE

Source dataset loaded: `ADSL.sas7bdat` (1,247 rows, 52 variables).

⚠️

Define-XML & SDTM Validation

⚠️ CHECK

SDTM variable `AEDECOD` present but not defined in active Define-XML v2.1. This may be a domain extension or version mismatch.

[Learn about domain extensions](#)

🔴

R Code Execution

✗ FAILED

Error at line 47 of `analysis.R`:

Error in `tapply(adsl_data$RACE, ...)` : RACE factor levels do not match expected controlled terminology. Expected: {'WHITE', 'BLACK', 'ASIAN', 'OTHER'} Found: {'White', 'Black', 'Asian', 'Other', 'Not Reported'}

[Show line 47](#)

🕒

Root Cause Analysis

→ SUGGESTED ACTION

Issue: Input data uses mixed-case RACE values, but your program expects uppercase to match CDISC controlled terminology.

Quick fixes: (1) Normalize values in your program with `toupper()`, or (2) Use the mapped RACE variable from the SDTM layer if available.

[Auto-apply normalization](#)

Study Protocol: `STUDY-2024-001`

Analysis Tag: `SAP-Section-5.1`

Audit Trail: `AT-00047392`

Ne

Rec

Add

Che

Verif

Defi

Bro

Revi

rule

[Documentation](#)

Similar Issues

3 other analyses faced RACE normalization issues in your project this month. View solutions.

[See Solutions](#)

Good UX - User is easily able to locate the error and check details that caused it leading to a quick discovery.

cdisc®

CDISC INDIA DAY
22 August, Chennai

Execution Failure — Diagnostic Chain

Execution Context

JOB ID

job-7f4a2b8c-e91d

USER

analyst@company.com

SUBMITTED

2026-08-10 14:32:17 UTC

R VERSION

4.3.1

CONTAINER

rwe-analysis:v2.1.4

ENVIRONMENT

production

Post scan - Tool suggests fix contextually and next steps saving a lot of time.

COMPLETE

Source dataset loaded: ADSL.sas7bdat (1,247 rows, 52 variables).

Define-XML & SDTM Validation

CHECK

SDTM variable AEDECOD present but not defined in active Define-XML v2.1. This may be a domain extension or version mismatch.

Learn about domain extensions

R Code Execution

FAILED

Error at line 47 of analysis.R:

Error in tapply(adsl_data\$RACE, ...) : RACE factor levels do not match expected controlled terminology. Expected: {'WHITE', 'BLACK', 'ASIAN', 'OTHER'} Found: {'White', 'Black', 'Asian', 'Other', 'Not Reported'}

Show line 47

Root Cause Analysis

SUGGESTED ACTION

Issue: Input data uses mixed-case RACE values, but your program expects uppercase to match CDISC controlled terminology.

Quick fixes: (1) Normalize values in your program with toupper(), or (2) Use the mapped RACE variable from the SDTM layer if available.

Auto-apply normalization

Next Steps

Recommended Fix

Add data standardization step before the failing tapply() call.

Apply Fix

Edit Program

Check Define-XML

Verify the AEDECOD extension is documented in your project's Define-XML. This is important for audit trail compliance.

View Define-XML

Browse Documentation

Review controlled terminology mappings and CDISC compliance rules specific to your study.

Documentation

Similar Issues

3 other analyses faced RACE normalization issues in your project this month. View solutions.

See Solutions

Study Protocol: STUDY-2024-001

Analysis Tag: SAP-Section-5.1

Audit Trail: AT-00047392

cdisc®

CDISC INDIA DAY
22 August, Chennai

Designing for the Agentic Future

Google Workspace has Gemini. Microsoft has Copilot. At Sycamore, we are building **S-CAFT™**.



AI only succeeds when users trust it. Good UX makes AI explainable.

Users should always understand WHY, WHAT, and WHERE before accepting an AI recommendation.

Showing a Visual mapping



Showing a Visual mapping and Contextual information
for users to reduce decision time



VARIABLE MAPPING <> PROGRAMMER VIEW X

AGE → DM.AGE

SOURCE · FORM ITEM

Form DM
OID LP.DM.AGE
Data Type integer
Codelist —

TARGET · SDTM VARIABLE

Domain DM
Variable AGE
Label Age
Core Exp

MAPPING FUNCTION

```
XCOPY(DM.AGE)
```

TEST DATA PREVIEW 3 SAMPLE ROWS

SUBJID	DM.AGE (RAW)	→ DM.AGE
S-001	42	42
S-002	67	67
S-003	29	29

Preview from most recent regen · [Run against latest snapshot](#)

DOWNSTREAM IMPACT 3 CONSUMERS

! ADSL ADAE

TLF T14.2 (Demographics)

ORIGIN & RATIONALE ● STANDARD

Direct standard mapping — AGE is a CDISC SDTM DM special-purpose variable.

REVIEW DECISION

Current: APPROVED

Add rationale / feedback (optional)

Approve Reject ⌵

Takeaways



CDISC standards define
how clinical data should look.

UX defines
how people experience those standards.

- AI-assisted clinical development
- Competitive advantage won't be smarter algorithms
- Designing systems that users actually **trust**.

Because the future of clinical data isn't only standards-driven. **It's human-centered.**

Thank you.

Questions & discussion



Narendra Parmar

 www.linkedin.com/in/nareniit/



Sycamore Clinical Agentic Framework™

SPD  | MDR  | DTM  | DSW  | SCE  | CDR 
PROTOCOL | METADATA | TRANSFORM | OPEN-SOURCE | ANALYSIS | REPOSITORY

Sycamore Informatics

 www.sycamoreinformatics.com



Scan & Connect !