

ML08: SPARC AI: A Native Metadata Driven Engine for End-to-End Clinical Trial Analysis

Authors:



Ashwin Kumar



Gourav Garg



Anil Mogha



DEDICATION

TRANSPARENCY

HONESTY

AGENDA



1

Limitations of current metadata and AI approaches

2

Solution : HYBRID AI

3

SPARC AI: A Practical Implementation

4

Architecture

5

Pilot Study results

6

Conclusion

LIMITATIONS OF CURRENT APPROACHES



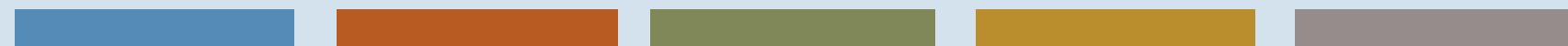
Metadata Only Systems

- Macros, Mapping systems
- Rigid

General Purpose LLMs

- Private code bases, limited training data. Not as good as Python or Javascript/React
- Lack study context
- Inconsistent outputs
- Explainability
- Hallucinations

Which LLM is better?



- Model capabilities are converging
- LLMs are being commoditized

Model	Organization	Global Average	Reasoning Average	Coding Average	Agentic Coding Average	Mathematics Average	Data Analysis Average	Language Average	IF Average
Claude 4.6 Sonnet Thinking Medium Effort	Anthropic	75.47	84.77	79.27	60.00	86.99	77.95	76.10	63.22
GPT-5.2 High	OpenAI	74.84	83.21	76.07	51.67	93.17	78.16	79.81	61.77
GPT-5.2 Codex	OpenAI	74.30	77.71	83.62	51.67	88.77	78.20	73.68	66.45
GPT-5.1 Codex Max High	OpenAI	73.98	83.65	80.68	53.33	83.22	70.12	76.48	70.38
Gemini 3 Pro Preview High	Google	73.39	77.42	74.60	55.00	81.84	74.39	84.62	65.85
Gemini 3 Flash Preview High	Google	72.40	74.55	73.90	40.00	84.17	74.77	84.56	74.86
GPT-5.1 High	OpenAI	72.04	78.79	72.49	53.33	86.90	69.61	79.26	63.90
GPT-5 Pro	OpenAI	70.48	81.69	72.11	51.67	86.17	57.04	80.69	63.96
Kimi K2.5 Thinking	Moonshot AI	69.07	75.96	77.86	48.33	84.87	61.36	77.67	57.41
GLM 5	Z.AI	68.85	69.11	73.64	55.00	83.46	67.90	77.53	55.33
GPT-5.1 Codex	OpenAI	68.61	81.98	71.78	53.33	79.58	60.75	69.48	63.39
Claude Sonnet 4.5 Thinking	Anthropic	68.19	77.59	80.36	53.33	79.31	56.97	76.45	53.35
GPT-5 Mini High	OpenAI	65.91	68.32	68.20	46.67	82.20	55.20	75.52	65.27
DeepSeek V3.2 Thinking	DeepSeek	62.20	77.17	64.62	40.00	85.03	50.00	70.41	48.19

Context is the king



- Given the lack of training data for llms, context becomes the key
- Study metadata, SAP, Specifications, Shells
- Programming libraries, functions, macros, etc

HYBRID AI: BRIDGING THE GAP



Why Hybrid?

- Metadata driven automation combined with constrained llm augmentation

Outcome

- Flexibility, control, auditability

SPARC AI: A Platform Approach



Why Platform?

- Metadata is native to the platform
- Access to study context
- Effectively use the llm code generation
- You get end to end traceability
- Reusability of the Programming libraries, reference codes, study metadata
- Human in the loop. Review and auditability

EXAMPLE SCREEN - 1

```
1  # =====
2  # TLF: Table 14.1.3.1
3  # Generated: 2026-01-15 17:05:05
4  # Study: Xanomeline-100-Spandex
5  # =====
6  # ===== LIBRARY_BLOCK_START =====
7  # Generated: 2026-01-15 17:05:05
8  # Library Loading code - Generated based on metadata
9  library(tern)
10 library(rtables)
11 library(dplyr)
12 library(haven)
13 # ===== LIBRARY_BLOCK_END =====
14 # ===== DATASET_BLOCK_START =====
15 # Generated: 2026-01-15 17:05:05
16 # Loading dataset from source
17 library(haven)
18 adsl <- read_sas("AdamDatasets/Outputs/adsl.sas7bdat")
19 # ===== DATASET_BLOCK_END =====
20 # ===== COLUMN_TEMPLATE_BLOCK_START =====
21 # Generated: 2026-01-15 17:05:05
22 # ColumnTemplate Name: Treatment Columns
23 column_template <- basic_table() %>%
24   split_cols_by("TRT01A", split_fun = keep_split_levels(c("Placebo", "Xanomeline Low Dose", "Xanomeline High Dose"))) %>%
25   add_overall_col("Total") %>%
26   add_colcounts()
27 # ===== COLUMN_TEMPLATE_BLOCK_END =====
28 # ===== LLM_BLOCK_START =====
29 # Generated: 2026-01-15 17:05:05
30 # Preprocessing
31 adsl <- adsl %>%
32   mutate(
33     AGE = as.numeric(AGE),
```

Metadata Generated

EXAMPLE SCREEN - 2

```
29 # Generated: 2026-01-15 17:05:05
30 # Preprocessing
31 adsl <- adsl %>%
32 mutate(
33   AGE = as.numeric(AGE),
34   AGEGR1 = factor(AGEGR1, levels = c("<65", "65-80", ">80")),
35   SEX = factor(SEX, levels = c("F", "M")),
36   RACE = factor(RACE),
37   WEIGHTBL = as.numeric(WEIGHTBL),
38   HEIGHTBL = as.numeric(HEIGHTBL)
39 ) %>%
40 filter(!is.na(TRT01A) & TRT01A != "")
41
42 # Analysis Implementation
43 lyt <- column_template %>%
44 analyze_vars(
45   vars = c("AGE", "AGEGR1", "SEX", "RACE", "WEIGHTBL", "HEIGHTBL"),
46   var_labels = c(
47     "Age (y)", "Age Group", "Sex", "Race",
48     "Baseline Weight", "Baseline Height"
49   )
50 )
51
52 # Build Table
53 tbl <- build_table(lyt, adsl)
54 # ===== LLM_BLOCK_END =====
55
56 # ===== TITLE_BLOCK_START =====
57 # Generated: 2026-01-15 17:05:05
58
59 # Main title
60 main_title(tbl) <- "Table 14.1.3.1"
61
```

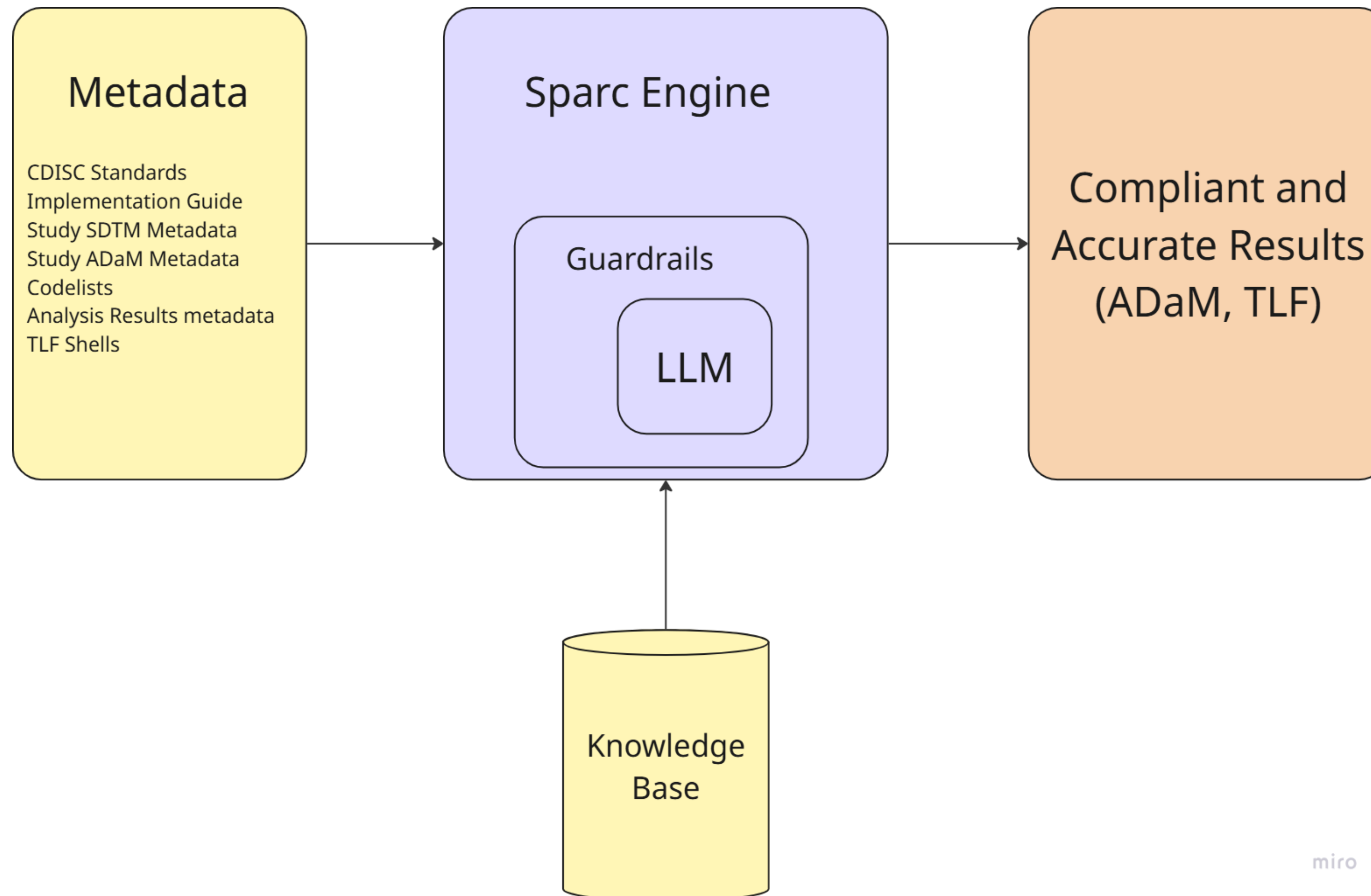
LLM Enhanced

EXAMPLE SCREEN - 3

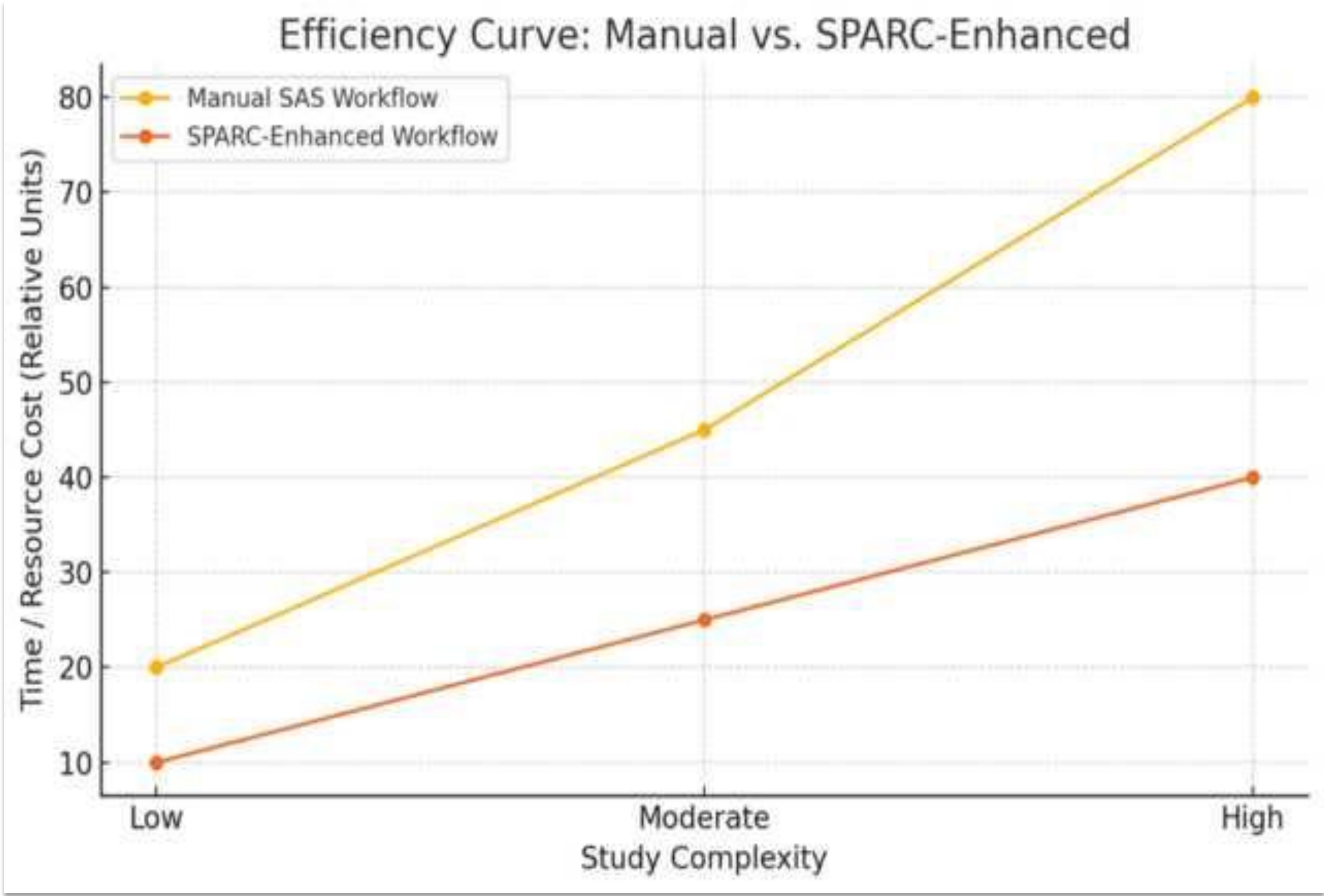
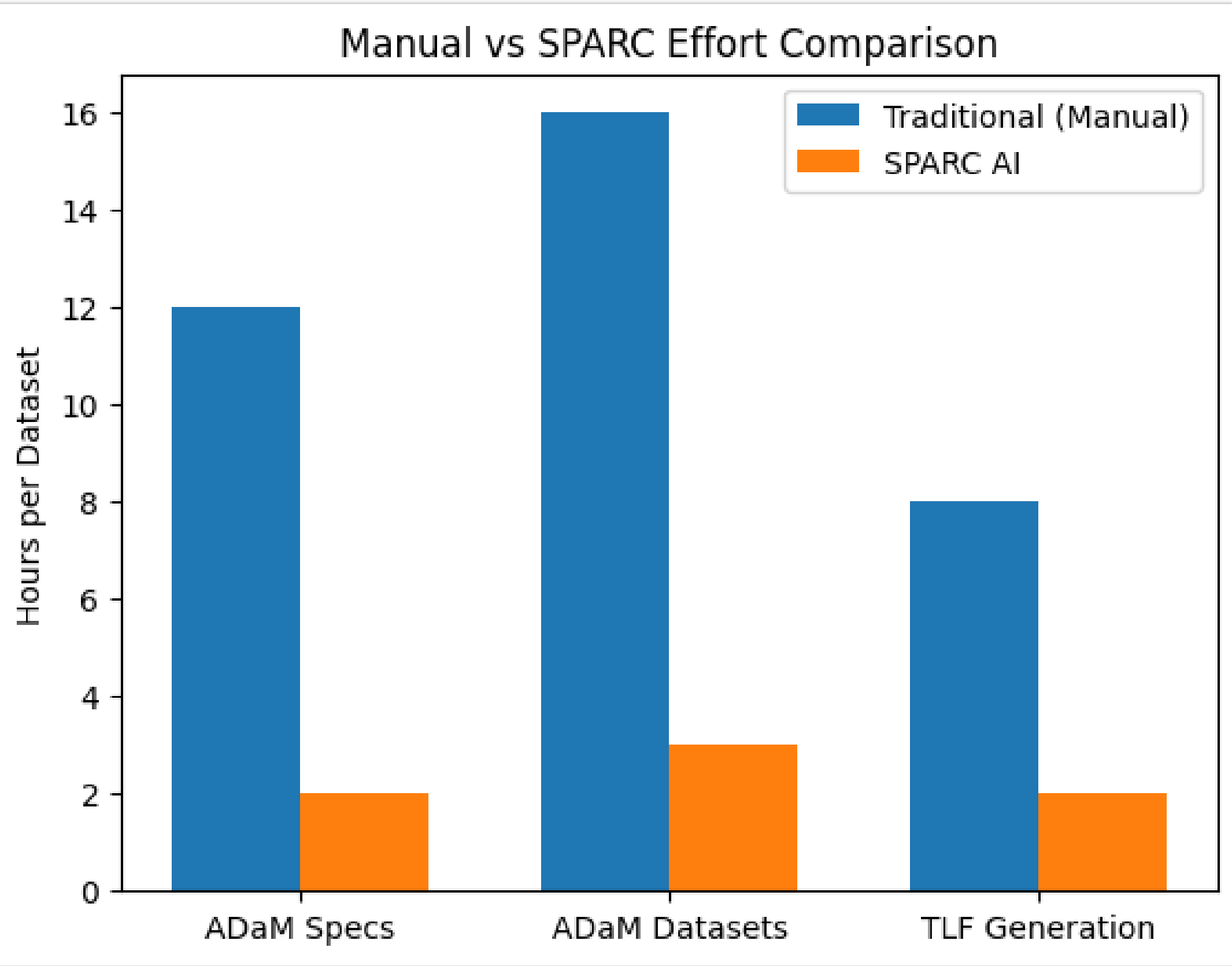
```
47 "Age (y)", "Age Group", "Sex", "Race",
48 "Baseline Weight", "Baseline Height"
49 )
50 )
51
52 # Build Table
53 tbl <- build_table(lyt, adsl)
54 # ===== LLM_BLOCK_END =====
55
56 # ===== TITLE_BLOCK_START =====
57 # Generated: 2026-01-15 17:05:05
58
59 # Main title
60 main_title(tbl) <- "Table 14.1.3.1"
61
62 # Subtitles
63 subtitles(tbl) <- c("Summary of Demographic and Baseline Characteristics")
64
65 # Print table
66 print(tbl)
67 # ===== TITLE_BLOCK_END =====
68
69 # ===== EXPORT_BLOCK_START =====
70 # Generated: 2026-01-15 17:05:05
71 # Export table output
72
73 export_as_rtf(tbl, 'TLFs/Outputs/t_table_14_1_3_1.rtf', landscape=FALSE)
74 # ===== EXPORT_BLOCK_END =====
75
76 # =====
77 # Code generation completed for TLF: Summary of Demographic and Baseline Characteristics
78 # =====
```

Metadata
Generated

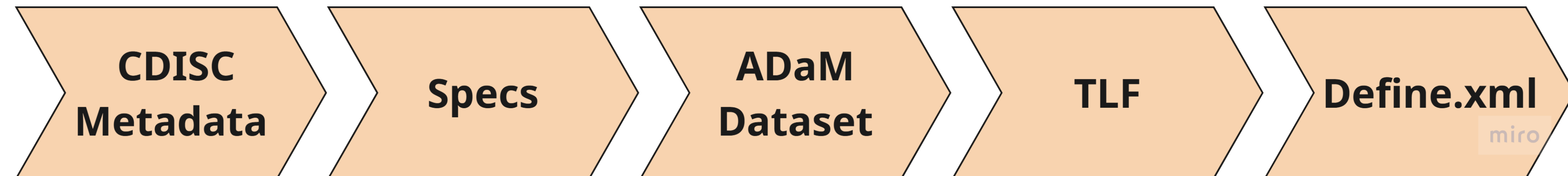
HOW IT WORKS



KPIs: MEASURING SUCCESS



THE CAPABILITIES



🏠 Sparc AI
Workspace Dev ⌵
Admin
A

< Back
Program: R ⌵
☀️ ▶ Run 📁 Save

```

62 # Dependencies: column_template_id
63 # Block Type: Column Template Configuration
64 # =====
65 # ColumnTemplate Configuration Code
66 # ColumnTemplate ID: ColTpl_1_CDISC_PILOT_STUDY_DEV_TREATMENT_COLUMN
67 # ColumnTemplate Name: Treatment Column
68 # Description: No description
69 # =====
70
71 # Create column template with predefined analysis group levels
72 column_template <- basic_table() %>%
73   split_cols_by("ARM", split_fun = keep_split_levels(c("Placebo", "Xanomeline Low Dose", "Xanomeline High Dose"))) %>%
74   add_overall_col("Total") %>%
75   add_colcounts()
76
77 # =====
78 # End ColumnTemplate Configuration
79 # =====
80
81 # ===== COLUMN_TEMPLATE_BLOCK_END =====
82

```

Chat with AI SparcLog

💬 Hello! I'm here to help you with the program. What would you like to discuss?

I want to split the columns again by SEX. can you modify the code

💬 To modify the code to split the columns by SEX in addition to ARM, you can add another `split_cols_by` call in the column template configuration. Here's how you can update the code:

```

1 # Create column template with predefined analysis gr
2 column_template <- basic_table() %>%
3   split_cols_by("ARM", split_fun = keep_split_level
4   split_cols_by("SEX", labels_var = "SEX", split_la
5   add_overall_col("Total") %>%
6   add_colcounts()
7
8 # Define layout and add analysis variables
          
```

Type your message... ▶

	Placebo (N=86)	Xanomeline Low Dose (N=84)	Xanomeline High Dose (N=84)	Total (N=254)
Age (y)				
n	86	84	84	254
Mean (SD)	75.2 (8.6)	75.7 (8.3)	74.4 (7.9)	75.1 (8.2)
Median	76.0	77.5	76.0	77.0
Min - Max	52.0 - 89.0	51.0 - 88.0	56.0 - 88.0	51.0 - 89.0
Age Group				
n	86	84	84	254
<65	14 (16.3%)	8 (9.5%)	11 (13.1%)	33 (13%)
65-80	42 (48.8%)	47 (56%)	55 (65.5%)	144 (56.7%)

THE CAPABILITIES



Reset Study



Workspace
Dev

Admin

A

>

← Back

TLF Metadata Configuration

Language: R

Configure metadata for TLF: Table 14.1.3.1

 AI Prediction

Use AI to predict appropriate metadata based on TLF shell content.

Predict Metadata

Metadata Configuration

Basic Information

Output Number

Table 14.1.3.1

Output Type

Table

Source Datasets

Source Datasets

adsl

Templates

Templates

Demographics Summary

Reference Codes

Select reference codes (programs or macros) that will be used for TLF generation. Only codes matching the selected language are shown.

Reference Codes

Analysis Configuration

Domain Objects

Analysis Set

Safety Population (AnIsSet_1_CDISC_PILOT_STUDY_DEV_SAFETY_...

Column Template

Treatment Column (CoITpl_1_CDISC_PILOT_STUDY_DEV_TREATME...

Titles

Title 1

Table 14.1.3.1

TLFs > Table 14.1.3.1

Summary of Demographic and Baseline Characteristics

TLF ID: Table 14.1.3.1 | Type: Table | Status: Dev

ShellHTML PreviewProgramOutputMetadata

HTML Preview

		Placebo (N=100)	Xanomeline Low Dose (N=100)	Xanomeline High Dose (N=100)	Total (N=300)
Age (y)	n	xx	xx	xx	xx.x
	Mean	xx.x	xx.x	xx.x	xx.x
	SD	x.xx	x.xx	x.xx	xx.x
	Median	xx.x	xx.x	xx.x	xx.x
	Min.	xx.x	xx.x	xx.x	xx.x
	Max.	xx.x	xx.x	xx.x	xx.x
Age Group	<65 yrs	xx (xx%)	xx (xx%)	xx (xx%)	xx (xx%)

Regulatory & Operational Impact

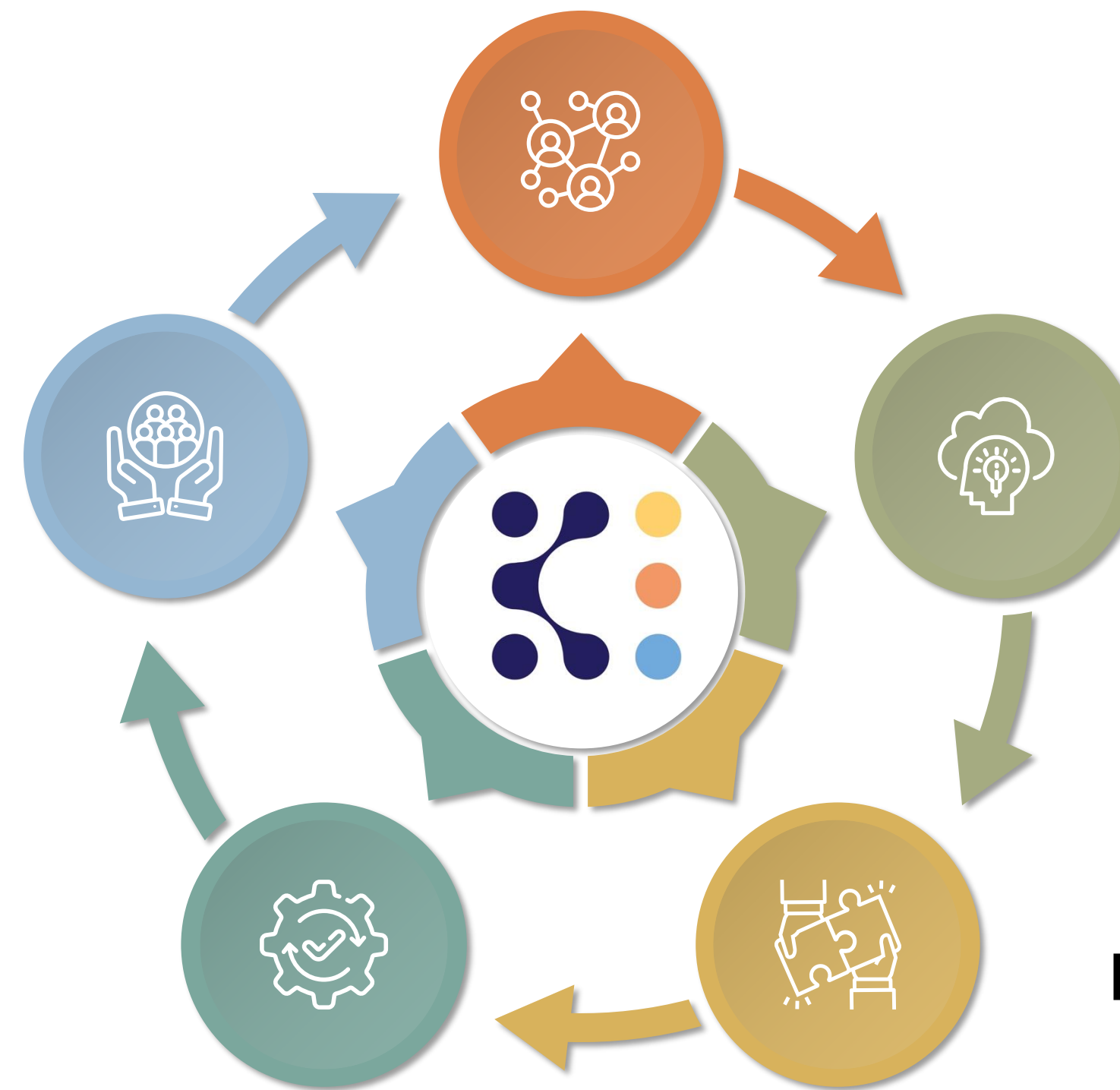
Scalable across studies

Faster trial close-out and reduced operational cost

Audit-ready outputs aligned with submission expectations

Deterministic, Compliant Outputs

Full end-to-end traceability across artifacts



Conclusion



- ✓ AI must be **metadata-native** to be regulator-ready
- ✓ Hybrid architectures balance flexibility with control
- ✓ Deterministic automation and LLMs are complementary
- ✓ Platforms like SPARC demonstrate a practical path forward

References

<https://k3-innovations.com/>
<https://sparcgen.com/>
<https://sparcgen.com/demo>



THANK YOU!!

CONTACT US

To know more on our solutions, visit : info@k3-innovations.com

Please contact +1-732-645-8366 101, or <https://www.k3-innovations.com/>

101 Morgan In, Suite 206, Plainsboro, New Jersey 08536