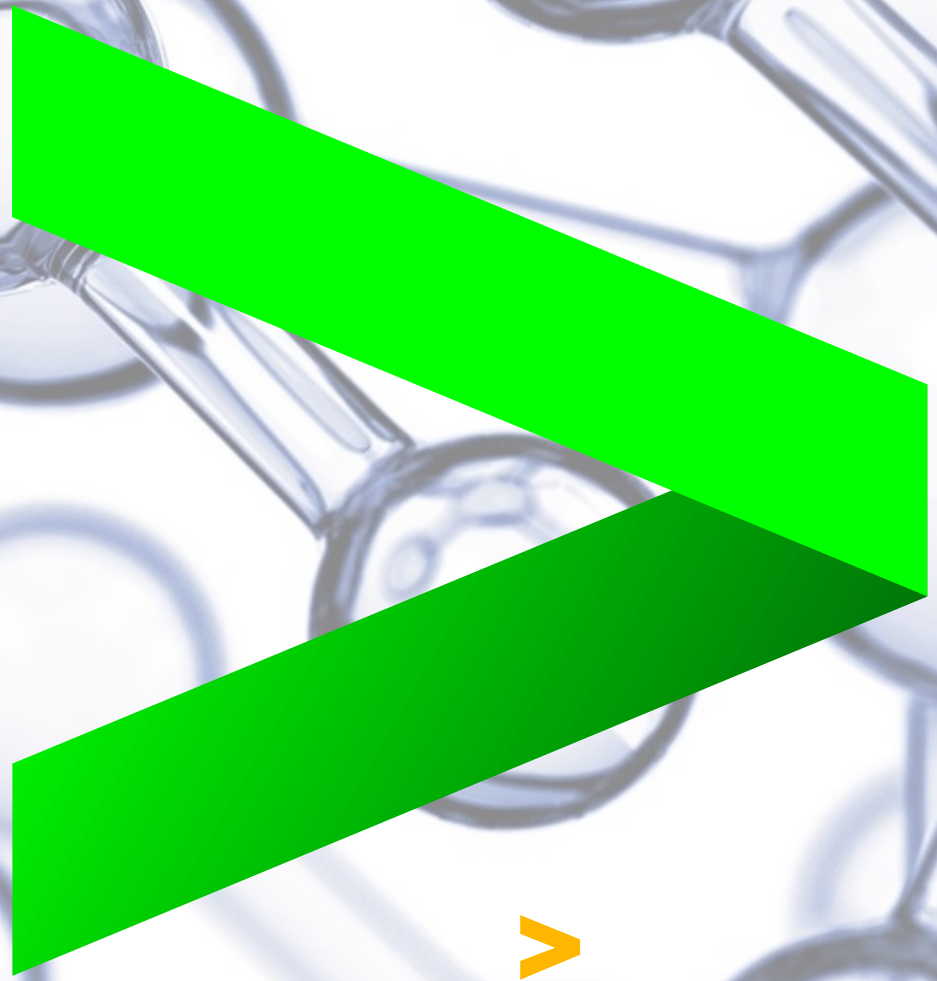
AUTOMATED METADATA-DRIVEN PROCESSING

PHUSE SINGLE DAY EVENT 2017: CROSSING THE TECHNICAL FRONTIER

Barry Cohen
Accenture Accelerated R&D Services

July 13, 2017



accenture

AGENDA

AUTOMATED METADATA-DRIVEN PROCESSING

BACKGROUND

What has brought us to this solution?

SOLUTION

What does it consist of?

BENEFITS, TAKEAWAYS

What might be the potential gains? Key messages.

LARGER CONTEXT

Accenture Intelligent Data Flow

BACKGROUND

- Industry Challenges

SOLUTION

What does it consist of?

BENEFITS, TAKEAWAYS

What might be the potential gains? Key messages.

LARGER CONTEXT

Accenture Intelligent Data Flow

CHALLENGES FACING THE INDUSTRY

Industry need

- Reduce amount of manual programming (and validation) needed to produce new artifacts in new study
  - Beyond what comes from re-use of standard artifacts
  - Reduce cycle time and labor

- Provide full, machine-readable, conveniently located record of all processing for a study
  - An historical, institutional record
  - What study artifacts were produced?
  - What was done to produce each study artifact?
  - Not in the program code; more convenient; more usable

SOLUTION

- Approach
- Components
- Usage

BACKGROUND

What has brought us to this solution?

BENEFITS, TAKEAWAYS

What might be the potential gains? Key messages.

LARGER CONTEXT

Accenture Intelligent Data Flow

APPROACH

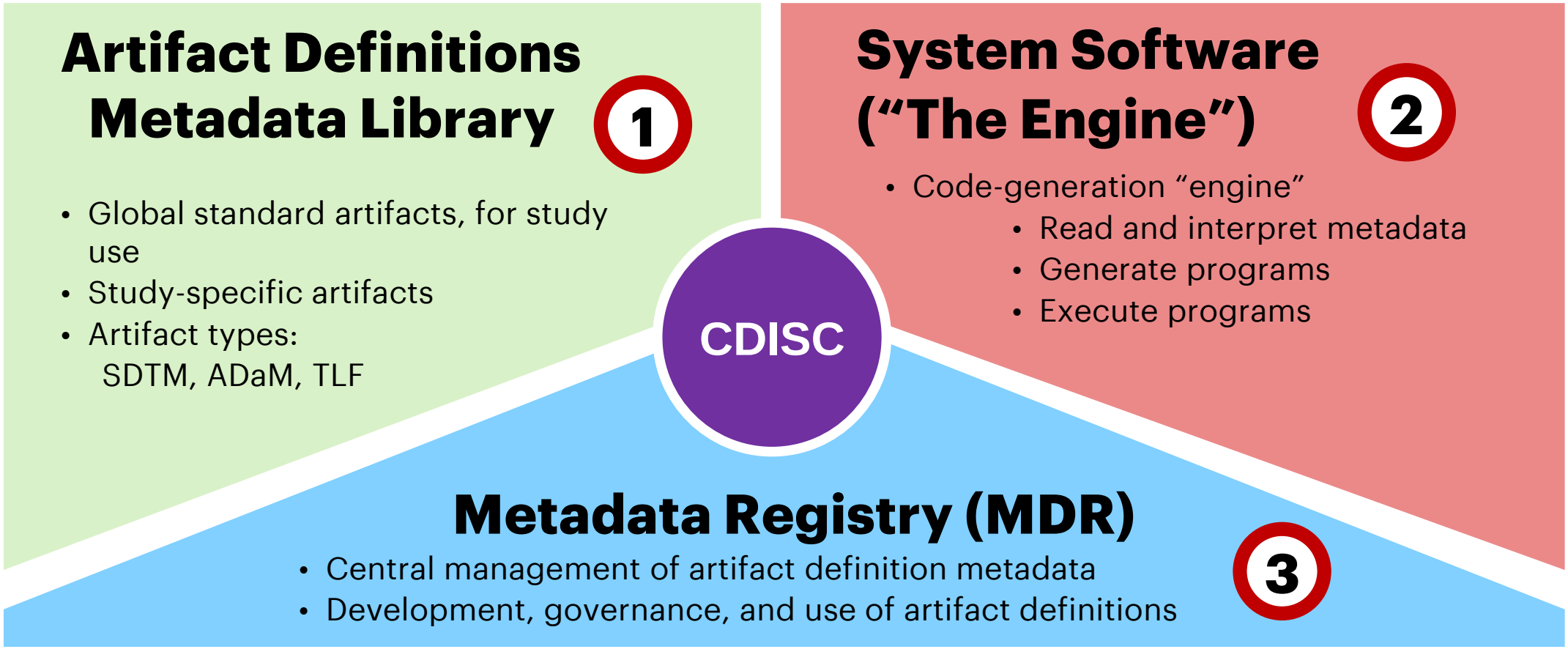
•Automated Metadata-Driven Processing (AMDP)

- Metadata – provides instructions to create study artifacts
- Code-generating software engine – auto-generates programs from metadata instructions, then executes programs to create artifacts

•Solution purpose

- Process efficiency – reduce manual programming (and validation)
- Process traceability –
  - Metadata describes the E2E process
  - Is a built-in machine-readable record of all processing and of relationships among study artifacts

HIGH-LEVEL COMPONENTS



★ **Artifact definitions are based upon CDISC standards**

HIGH-LEVEL COMPONENTS

1

Artifact Definitions Metadata Library

- *Definitions for artifacts produced for a clinical trial*
- *Data artifacts: Tabulation (SDTM) and Analysis (ADaM)*
- *Report artifacts: Analysis Report (TLF)*
- *Global standard artifact definitions*
- *Study-specific artifact definitions*

ARTIFACT METADATA OVERVIEW

Artifact-Level

- Components of artifact definition
  - Structure – for datasets; for tables/listings
  - Process
    - Transformation, derivation, computation
    - Data joins
    - Sequence of processing
- Forms of artifact definition
  - Human-consumable – structure and process; system-agnostic
  - Machine-consumable – structure and process; system-specific

Study-Level

- Plan for artifacts creation
  - What artifacts to create, in what order

HIGH-LEVEL COMPONENTS

2

System Software – "The Engine"

- *A set of programs designed to:*
  - *Read, interpret, validate metadata*
  - *Generate artifact-creation programs*
  - *Execute artifact-creation programs*
  - *Utilize pre-written utility macros in program generation*

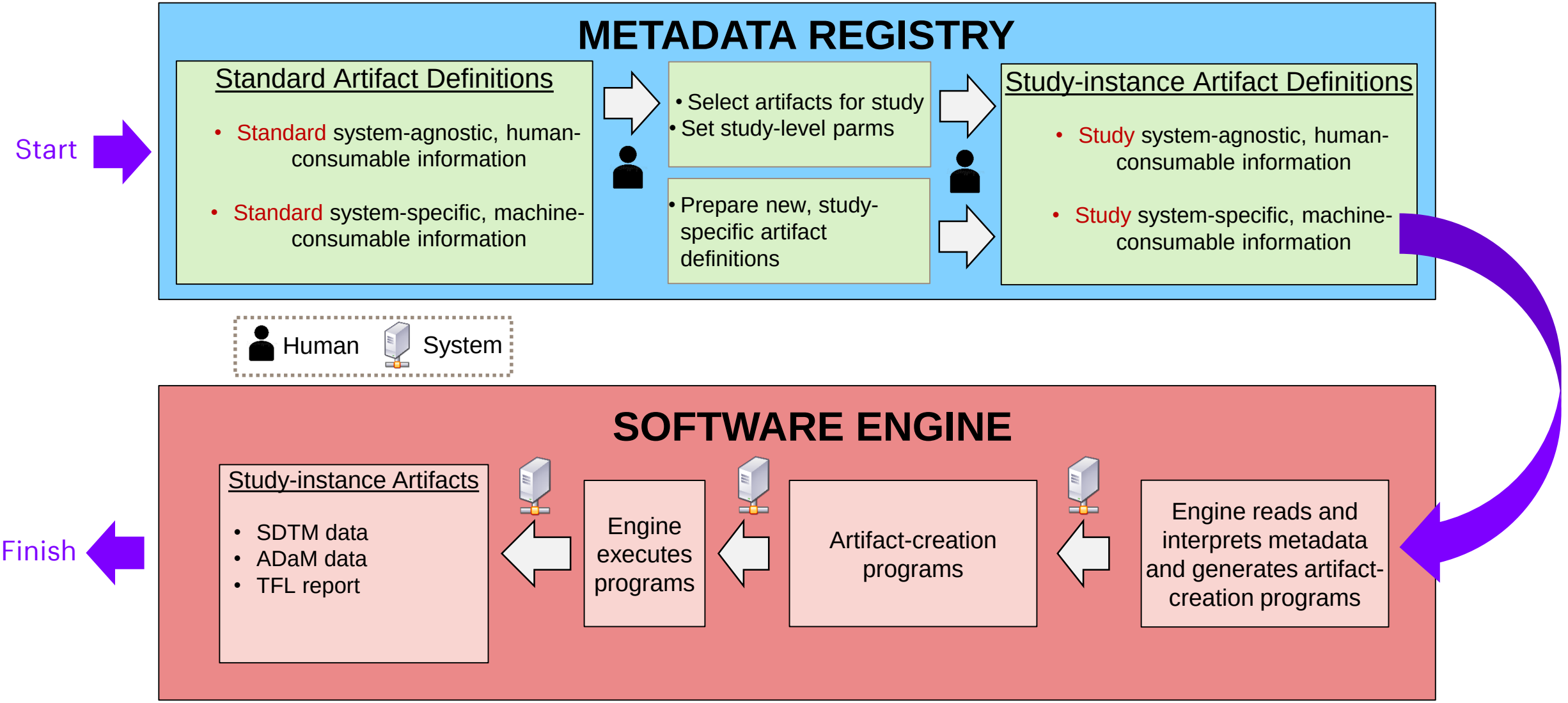
HIGH-LEVEL COMPONENTS

3

Metadata Registry

- *Create, maintain, govern artifact definition metadata*
- *Version control, audit trail, access control*
- *Governance workflows*
- *Standards and study definitions repositories*
- *Accessible by the System Software Engine*

PROCESS FLOW FOR STUDY AUTOMATION



STUDY ARTIFACT CREATION SCENARIOS

	Definition exists in library	Definition does not exist in library		
	(1) Create existing standard artifact as-is	**(2) Create new similar artifact from standard definition w/o programming**	**(3) Create new similar artifact from standard definition w/ some programming**	**(4) Create new dissimilar artifact by authoring full new program**
Description	• Copy and use std. metadata as-is (set macro parm values) • Use utility macros as-is • No programming	• Copy and edit std. metadata • Re-use existing utility macros • Std. metadata and utility macros are building blocks • No programming	• Copy and edit std. metadata • Re-use existing utility macros • Plus, author new utility macro for balance of artifact	• Author new custom program to fully create the artifact • Author new, simple metadata to execute new specific program from inside the system
Metadata-Driven	• Fully	• Fully	• Partial	• No, but Software Engine will run the custom program
Examples	• SDTM from std. CRFs • TLF from std. layout	• Custom SDTM domain • Some ADaM (e.g., add new categorical var from continuous using existing utility macro)	• Some ADaM (e.g., add unique analysis vars; add new rows)	• Analysis dataset with unique structure • TFL w/ unique layout
Est amount in study*	• 60% +/- of artifacts	• 20% +/- of artifacts	• 10% +/- of artifacts	• 10% +/- of artifacts

*Accenture industry estimate across all artifact types. Actual amount will vary by company and by artifact type.

BENEFITS, TAKEAWAYS

- Efficiency; Traceability
- Takeaways

BACKGROUND

What has brought us to this solution?

SOLUTION

What does it consist of?

LARGER CONTEXT

Accenture Intelligent Data Flow

EFFICIENCY GAIN AREAS

	(1) Create existing standard artifact as-is	(2) Create new similar artifact from standard definition w/o programming	(3) Create new similar artifact from standard definition w/ some programming	(4) Create new dissimilar artifact by authoring full new program
Conventional Approach	• Set up env. and call artifact macros via written programs • Jr. programmer	• Copy/edit existing program, or author new program • Jr/Sr. programmer	• Copy/edit existing prog. • New program for new part of artifact • Jr/Sr. programmer	• Author new program • Sr. programmer
AMDP Approach	• Set up env. and call artifact macros via metadata • Jr. metadata curator	• Copy/edit existing metadata • Jr/Sr. metadata curator	• Copy/edit existing mdata • Author macro for new part of artifact • Jr/Sr. metadata curator • Jr/Sr. programmer	• Author new program, w/ more macro library use • Sr. programmer • Jr/Sr. metadata curator
Efficiency Gain Areas with AMDP	• Less experienced resource • Easier to engage metadata than programs • Less QC b/c no double programming • Preserve processing record	• Less experienced resource • Easier to engage metadata than write program • Less QC b/c no double programming • Preserve processing record	• Less experienced resource • Easier to engage metadata than write program • Less QC (only double program new part) • Preserve processing record	• Faster programming b/c more likely to know and use utility macros • Preserve processing record
Relative AMDP gain	• Mid. efficiency gain	• Max. efficiency gain	• Mid. efficiency gain	• Min. efficiency gain

- Amount of saving per study is function of how many artifacts are in each scenario
- Utility macro library is enhanced over time, as more artifacts encountered; thus, relative number of artifacts in #1 & #2 increases, and in #3 & #4 decreases, over time, thereby increasing gain.

KEY TECHNICAL TAKEAWAYS

- **System validation and artifact definition validation are different**
  - System (engine) validation – IT-centric issue
  - Std. artifact definition validation – std.'s team centric issue
  - Confusing because artifact definitions needed to validate system
- **Engine must generate the complete program, else lose efficiency**
  - Lock the generated program from editing
- **Avoid much executable code in the metadata**
  - Else, just moving manual programming to new place, and still need program validation
  - Keys: (1) robust utility macro library; (2) strong training

KEY TECHNICAL TAKEAWAYS

- **TLFs are different, harder to automate, but key to success**
  - Different than data creation - not transforming or deriving data
  - Must address report structure in generalized, standard way
  - Perhaps largest single challenge of the system development
  - Largest volume of artifacts, so largest payoff from automation

KEY METADATA TAKEAWAYS

- **Standard definitions are hard to achieve, especially for ADaM & TLF**
  - Special challenge if AMDP project also bringing higher level of standardization to the organization, in general

- **Risk-based validation for study artifact creation**
  - Structure it around artifact creation scenarios
  - Deeper validation needed as move from:
    - Use existing to prepare new metadata
    - Simple to complex metadata
    - No program code to program code
    - Limited program code to much program code

KEY EFFICIENCY TAKEAWAY

- **How much efficiency gain per study?**
  - **Factors**:
    - Size of standard artifacts library
    - Robustness of utility macro library
    - Mix of standard vs. new artifacts in a study
    - Mix of new similar vs. new dissimilar artifacts in a study
    - Staff proficiency
  - **Accenture experience is too new for empirical data**
    - Expecting 30%+, and see it going higher

KEY STRATEGIC TAKEAWAY

- **Not for the feint of heart**
  - **Major effort to:**
    - Implement a solution that can generate 100% of needed program for an artifact
    - Build a library of standard artifact definitions
    - Change the way the organization approaches business
    - Obtain patience and budget from sr. leadership, given long path to full payoff
  - **Others have tried this approach**
    - Mixed results across industry
  - **Accenture seeing success and substantial value**
    - Intend full implementation for both our services and product

LARGER CONTEXT

- Accenture Intelligent Data Flow

BACKGROUND

What has brought us to this solution?

SOLUTION

What does it consist of?

BENEFITS, TAKEAWAYS

What might be the potential gains? Key messages.

INTELLIGENT DATA FLOW

Point today **Metadata provides intelligence and automation. Automated metadata-driven processing is a key component, and is intended for Accenture service delivery and product.**