Building Quality and Efficiency into Biometrics Tools and Processes

Presented by Linda Collins and David Brega

Data and documentation standards are more essential and complex than ever

- Traditional data pathway
  CRF -> Raw Data -> SDTM -> ADaM -> TFL -> Study Report.
  Documentation is tacked on at the end.

- Traditional methods: write programs and documentation without programmatic connections. This has important drawbacks.
  - Hard to verify, labor intensive, limited reusability, no management of higher-level standard content, etc.

What's shifted? What hasn't?

- Standards are being required/enforced (FDA Guidance Dec 2014).
- Requirements are more complex
  (new SDTM domains, TAUGs, ADaM structures).
- Studies are more complex.
- Standardized structure and terminology exist at multiple levels.
- Expectations about documentation are ratcheting up.
  Define 2.0 is more robust, richer in capabilities, but harder to implement.

- But the need to express study-specific data points will never go away.
- Making your argument clear to a reviewer is *the whole point*.

What are your needs?

- Small pharma needs to crank out a few studies of superior quality while figuring out how to do it.
- Mid-sized pharma needs to expand its throughput from 3-6 studies per year to 150-to-450 studies per year, and eventually its large pharma goal.
- Large pharma has to produce 1,200 to 1,600 studies per year while maintaining end-to-end work product dependencies.
- Full-service CROs have an altogether different problem set. (Quick response, no control of user requirements, hard to leverage toolsets)

All clinical data processing is part of a pathway of data flow

- Some parts may be manual, but they are still part of the process.
- Anything that can be captured in metadata can be subjected to programmatic management and checking.
- Moving tasks from code to metadata can make life better

Needed: a system of processes, and metadata to implement them

- Requirements for a system to manage elements of a standard process could include:
  - Elements of data to be managed
  - Mechanics of managing metadata elements
  - Programmed and manual relationships between ‘moving parts’
  - Control of execution and checking
  - Granularity and versioning
  - Facilities and features needed
  - Governance: who decides?
- Not everything needs to be done at once. Even small improvements can pay big dividends.

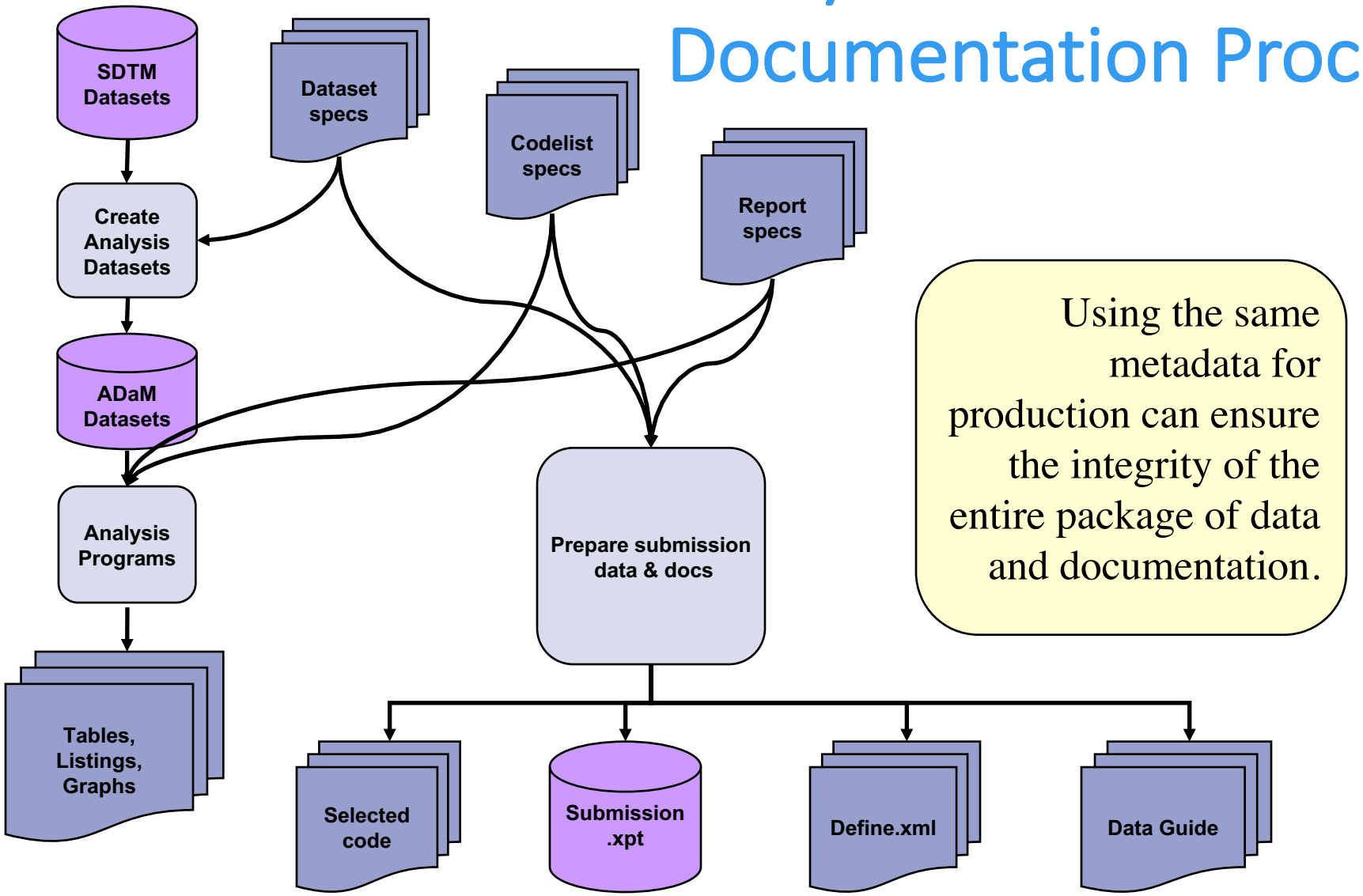
What elements should be managed?

- Data capture modules
- Structural: Dataset names, variable names, codelists, derivation rules, source -> target relationships, references to external documentation
- Transformations: descriptions of data manipulation.
- TLF elements: layouts, input definition (datasets and variables), titles and footnotes, statistical modules.
- Submission documentation: define and reviewer guide content

What goes into the ‘managed elements’?

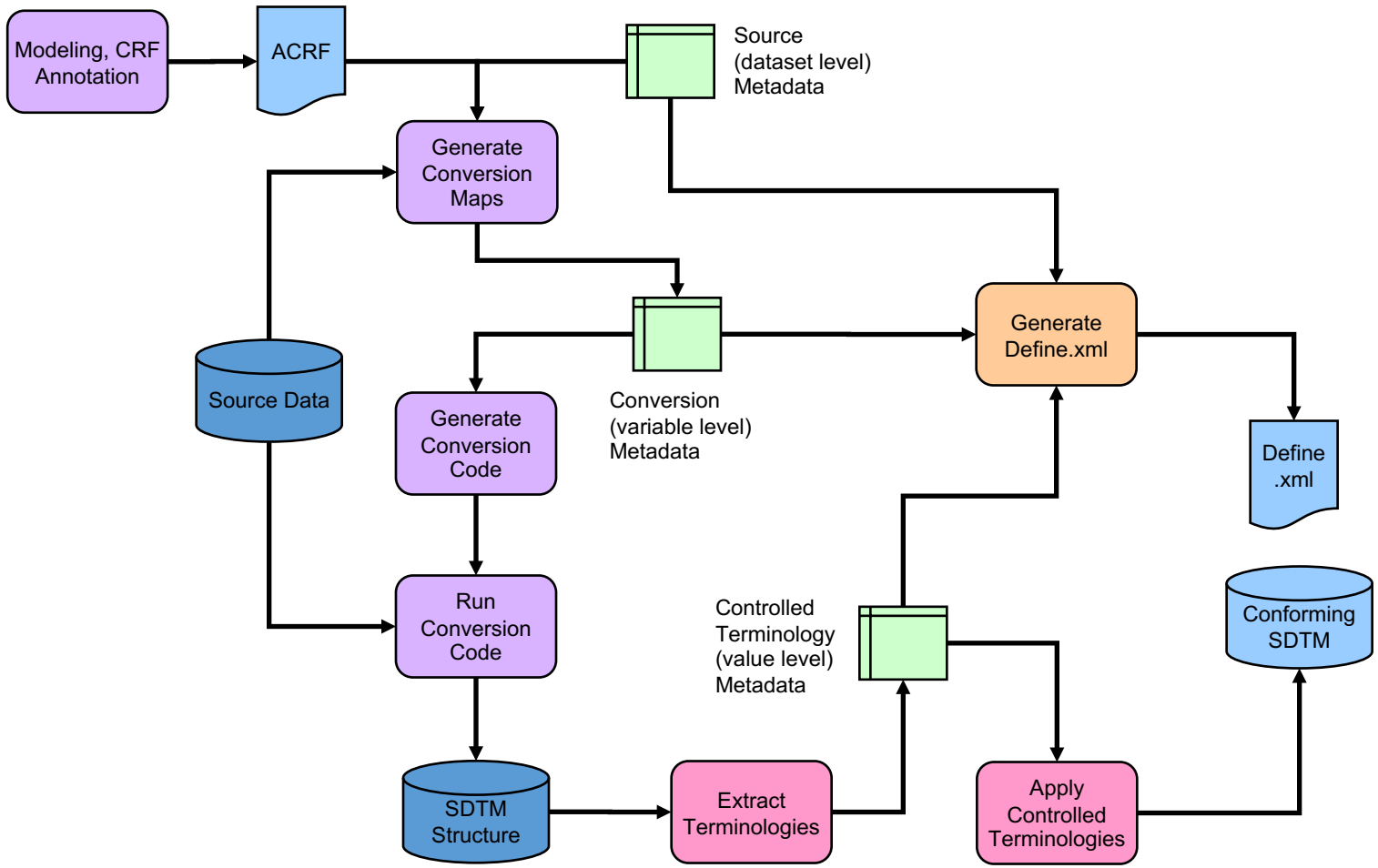
- Published standards give us a framework, but they are too general to serve all purposes.
- You need defined rules for how to create each work product, what it should look like, and what it should convey. Detailed SOPs and work instructions can ensure consistency.
- These rules should NOT accommodate for every possible study configuration, or be updated for each individual study.
  - The resulting metadata would be large, unwieldy, and impossible to version control.

Analysis and Documentation Process



Using the same metadata for production can ensure the integrity of the entire package of data and documentation.

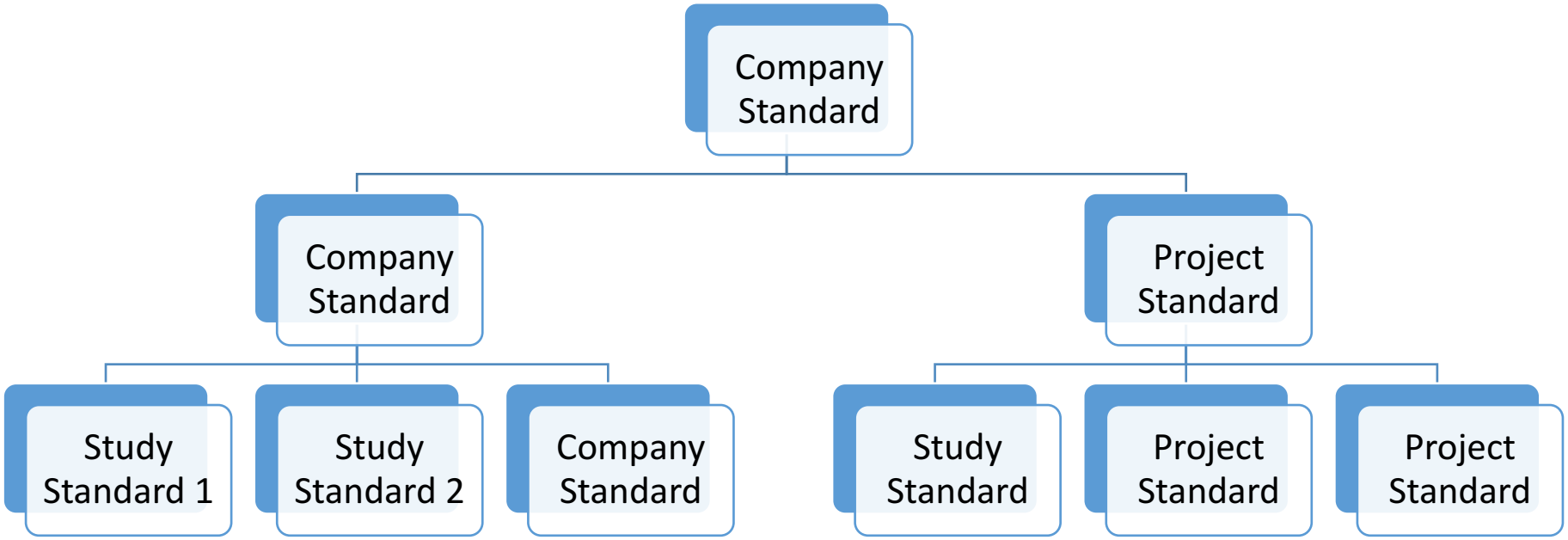
SDTM Process Overview



What's the plan for control of the execution and checking of a process?

- Did everything run?
- In the right order?
- Correctly? Without snags caused by new or problematic data?
- How do you know an entire process is done?

Use hierarchical metadata structures



- Different organizations will need different numbers of layers
- This gives you the ability to add specificity, or deviate when necessary
- Therapeutic areas may be needed

What's the right granularity and versioning for 'managed elements'?

- How many layers need to be managed
  (public, company, therapeutic area, protocol, analysis)?
- How do public elements (e.g. CDISC controlled terminology) or higher-level (e.g. company standards) get brought into a project?
  Is it a static copy or used by reference?
- Do multiple versions need to be maintained?
  Active simultaneously?
  How do you track what was used?

What are the relationships between the 'moving parts'?

- What is the relationship between 'managed elements' and their outputs? If it's possible to clearly define it, can it be programmed in a consistent way?
- All metadata must have a defined structure.
  Think about a structure that can be used by all parts of the process that it can impact. For example, if a certain variable has a codelist, is that codelist used in tables?
  In define.xml documentation? How is that codelist connected to higher-level controlled terminology?

Where and how do you house these elements?

- Excel, RDF, XML
- Code libraries
- Metadata repository

- What structures do you need in the metadata to support all intended uses?
- Do these structures support programmed uses?

Who decides?

- Process for making decisions about usage of common elements is essential.

  Without a governance structure, every task will remain a one-off process.
  Who determines the elements to be used and how they are used?
  Who gets to identify and approve extensions? Exceptions?

- The most successful implementations we've seen take the role of standards 'gatekeeper' seriously and give them authority to do the job.

Ensuring the integrity of an SDTM conversion: some needed diagnostics

- Accounting for all input variables from all data sources; documenting why certain variables are not used
- Datasets are consistent with relevant CDISC standards:
  Naming conventions and attributes
  Populated per rules in standards: e.g.,
  - Required variables, variables that should not be empty, variables that should have 1:1 relationships, etc.
- Controlled terminology is applied properly
- Derivations are correctly performed

Ensuring the integrity of a submission package: some needed diagnostics

- Define.xml (and associated docs) ***match dataset content***:
  Contains all (and only) existing datasets and variables
  Match on all attributes
  Content matches associated codelists

- Define.xml (and associated docs) are ***technically conforming***:
  Well-formed XML
  Render properly using style sheet
  All navigation and links work correctly
  Legitimate values for non-displayed variables (mandatory, signif digits)

- In Define.xml 2.0 the 'package' can include analysis results metadata
  This will involve ***new consistency rules***

How do you know the whole process works properly?

- Validation
  - In-house software development needs an SDLC procedure.
  - Software validation is an underappreciated need in many shops.
  - Even COTS products need PQ (production qualification).
- Documentation and training
  - The best system in the world can be defeated if people aren't part of the equation.
  - No software is immune to abuse.
  - Programmed checks are fine, but people have to be trained to pay attention to them. Someone has to tell them what to do with diagnostics!

Where to start?

- Decide which problems you need to solve.
- Look at how these problems are being addressed currently.
  (Candidates for metadata or parameters?)
- Evaluate the necessary tools.
  Do make-vs.-buy evaluations.
- Prioritize upgrades to your processes.
- Look at your governance process.
  (Is there a good gatekeeper available?)

Step 1: Make a project plan.

A project plan needs to include the elements to be automated, interim milestones, timelines, resources, plans to roll out new functionality, and budget.

- Implementation Master Plan
  - Define subtasks, deliverables, resources, timing
- Governance
  - Define things to govern and structures to hold them
  - Set up a curation and escalation process for SDTM, ADaM, TFLs and source code.
  - Some analysis elements will remain outside standards. Decide on a process for this.
- Metadata content
  - Continue development of initial content
  - Develop procedural and automated uses of content
- Internal software development
  - Identify resources for automating some steps
  - Use governed metadata and code as part of the data flow
- COTS evaluation
  - Consider tools such as MDR, support for versioning

Don’t Panic!

- It’s a big job.
- It can be done.
- It’s worth it.

Contact Us

- Linda Collins, lcollins@pharmastat.com
- David Brega, dbrega@pharmastat.com