



GUI for CSR reporting: R you ready to Shine with SAS?

Chris Ridley and Matthew Brierley

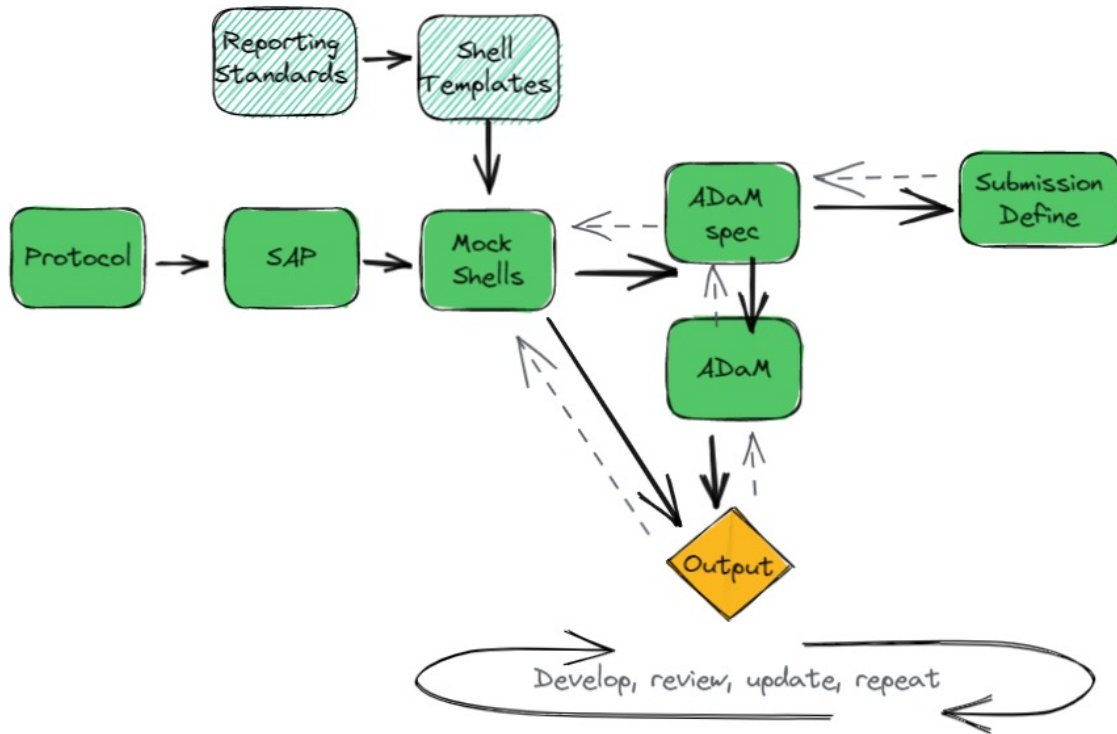
✉ chris.ridley@ext.idorsia.com

✉ matthew.brierley@ext.idorsia.com



Problem statement

Generating regulatory reports for a CSR is inefficient!



- Manual processes operating in silos
- No holistic overview of what data is used where
- Poor standards adoption
- ADaM does not always fully accommodate output
- High-quality define.xml only produced at time of submission
- Metadata not leveraged during both CSR design and reporting stage

New tools and skills in the industry has led to different teams exploring new solutions for different needs.....
... we don't look for a new need, we want to add **more value** to **existing** CSR output **requirements**

A better way...?

We have rich data standards within CDISC, we should leverage them!



Define
metadata
up front

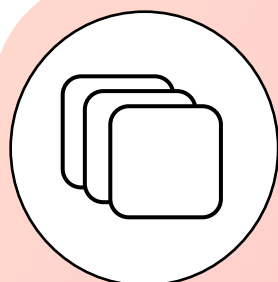
Process change: Design
ADaM define.xml in parallel
with mock shells

P21

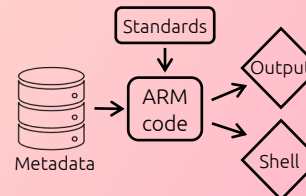
Leverage ADaM
metadata from
Pinnacle 21

ARM+

Build metadata link
between Output
and ADaM



Generate
programs from
metadata



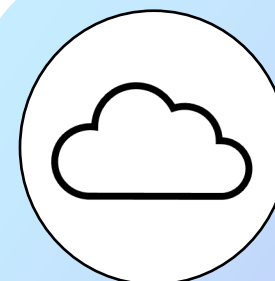
Code will **generate**
shells to match reporting
standards



ADaMs built; same code will
create the **real outputs**



We now see the datasets
and variables used in our
reporting

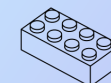


Central
Platform

Create **single application** to
bring the concept together



Review data / metadata



Design shells / output



Generate & execute code



View results



Version control

Starting Point

ARM

ARM_VARIABLE_

KEYSEQ_

This should start at 1 and then run sequentially within each output. In most cases, the variable with lowest key seq is presented first

Enter the full variable address. E.g. ADSL.ACOUNTRY

ADSLAGE

ADSLAGEGR1N

ADSLAGEGR2N

ADSLASEXN

ADSLARACEN

ADSLAETHNIC

ADSL.HGTSC

ADSL.WGTSC

ADSL.BMISC

ADSL.BMISCCTN

ADSL.REGION1N

ACT246475_ACS / ID076A201 CSR

Table 15.1.4.1-1.1 - Demographics characteristics
Full Analysis Set

Page 1 of 1

Variable Statistic	ACT-246475 8 mg test N=44	ACT-246475 16 mg test N=44	Placebo test N=38
Sex [n(%)]			
n	44	44	38
Male	31 (70.5)	35 (79.5)	31 (81.6)
Female	13 (29.5)	9 (20.5)	7 (18.4)
Age (years), EudraCT [n(%)]			
n	44	44	38
< 18	0	0	0
18 - < 65	21 (47.7)	19 (43.2)	18 (47.4)
>= 65 - < 85	23 (52.3)	25 (56.8)	20 (52.6)
>= 85 and over	0	0	0
Age (years) [n(%)]			
n	44	44	38
< 55	7 (15.9)	7 (15.9)	4 (10.5)
>= 55	37 (84.1)	37 (84.1)	34 (89.5)
Race [n(%)]			
n	43	43	36
Black or African American	7 (16.3)	2 (4.7)	6 (16.7)
American Indian or Alaska Native	0	0	0
Native Hawaiian or other Pacific Islander	0	0	0
Asian	2 (4.7)	1 (2.3)	1 (2.8)
White	33 (76.7)	40 (93.0)	29 (80.6)
Missing	1	1	2

Program: T_DM_01_FAS.sas.
Output generated: 2018-08-21 10:19 ('GMT+02:00'). Database <CUT-OFF/LOCK> date: <YYYY-MM-DD>.

_VAR_MACRO_

signs (EQ, NE etc) and associated macro - output level

m_mixed

m_mixed

m_mixed

m_mixed

m_mixed

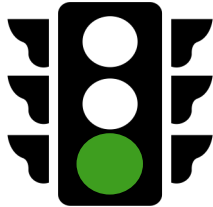
m_mixed

m_mixed

m_mixed

m_mixed

m_mixed



Project launch!

2018 ...



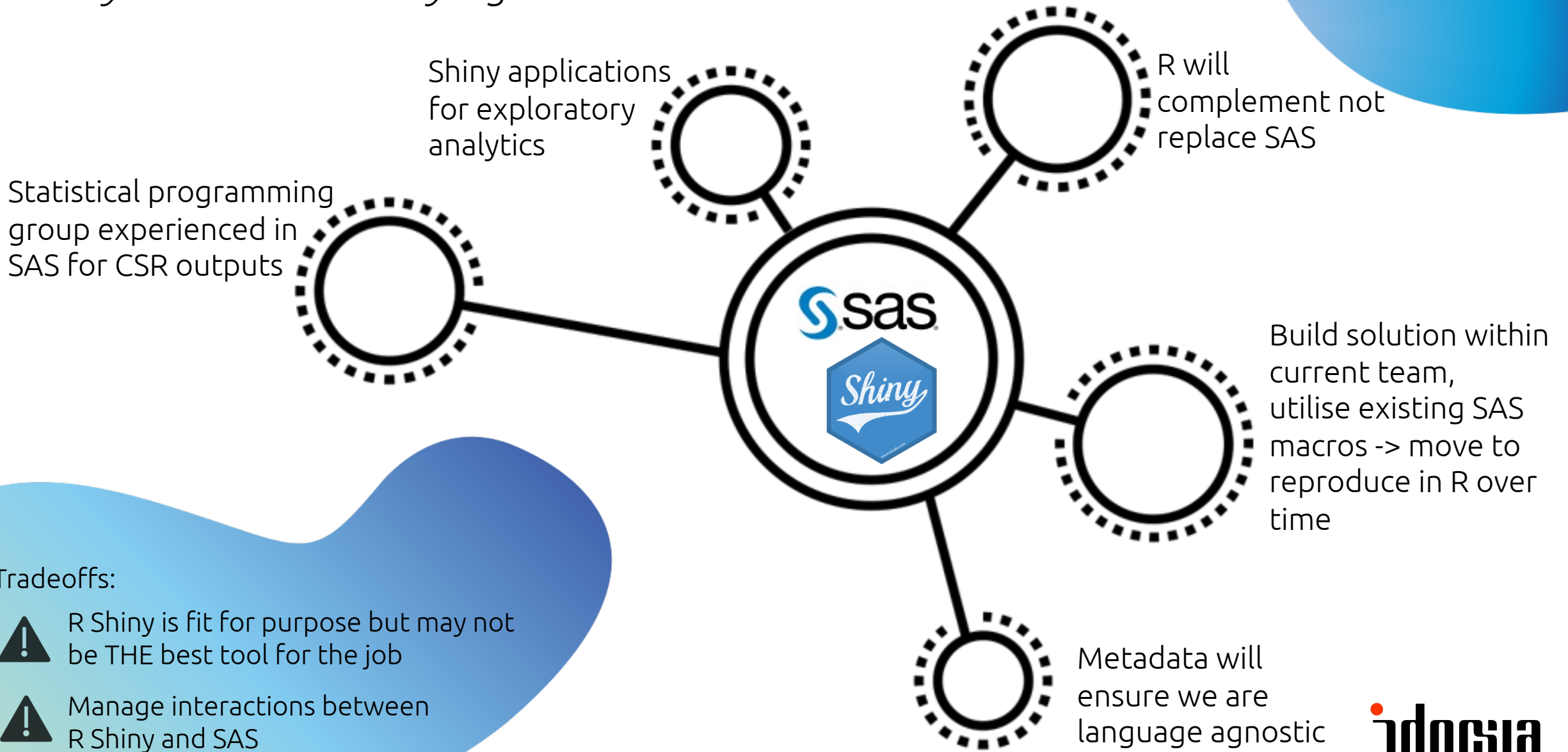
R / RShiny uptake

... 2020



Looking in the toolbox

Why use SAS and R Shiny together?



Tradeoffs:



R Shiny is fit for purpose but may not be THE best tool for the job



Manage interactions between R Shiny and SAS

Demonstration



Project Status

Where are we today?

Rapidly build shell and/or output from scratch

Execute and view SAS outputs in R Shiny on the fly

ADaM and outputs are connected by ARM+

Store versions of specifications

View and search metadata

Output schema established

Where are we going?

Multiple TOCs per study

Template library of standard outputs

Further integration with programming environment + P21 API

Extend R functionality -> support validation

Link exploratory Shiny apps to CSR metadata to ensure consistency

Dashboard summaries of our metadata



Acknowledgements

- John Coene
- Pedro de Silva
- Jonathan Reid
- Scott Pain
- Francesca Massa-Rolandino





Thank you

