

Automated self-documenting Programs, Automated Table Shells, Landing Zones and the Importance of Standards

PhUSE SDE Basel, 16MAY2017

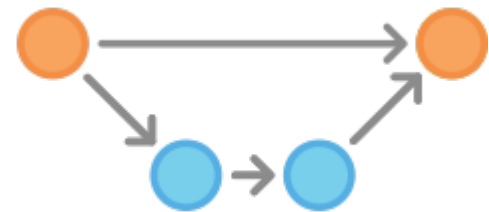


Christoph Ziegler, Roche



Agenda

- Background (Automation and Workflow Optimization)
- Automated self-documenting Programs
- Automated Output / Table Shells
- Study / Project Landing Zones
- The Importance of Standards
- Summary & Outlook & Further Thoughts
- Q & A



Background – Workflow Automation and Optimization of Processes

- „Hot“ topic within the entire industry
- Money- and time-saving is becoming more and more important
- Agility / flexibility / sense of urgency often lacking
- Competitors & specialized companies in rare diseases
- Being first in the market & biosimilars / generics
- Growing data & data sources
- Growing complexity of data (Biomarker, genetic data,...)
- Healthcare is about to change (Personalized healthcare)
- Automation and optimization is needed, **also from statistical programming**



Background – Workflow Automation and Optimization of Processes

- What in general can statistical programming do?
 - Optimize and automate things wherever possible
 - Contribute to the ultimate goal of the function / organization
 - Live the statistical programming analyst mindset
 - Speak up
 - Avoid overhead
 - Contribute to agility and speed
 - Focus efforts on the right things
 - Work closely and effective with contingent resources & stakeholders
 - Work on project management skills
 - Effective resourcing
 - „right work for the right people“



Automated self-documenting Programs



Problem

- It is inefficient to document program derivations in program comments / the program header etc. and at the same time in external documents like e.g. MS Excel & MS Word
- Health Authorities request readable / executable code
- A program will be changed, most probably by someone else...

Solution

- Write and do programming specifications in the actual program code that is written
- An automated program specification file can be created at run-time of the program
 - specifications are always up-to-date
 - good commented programs
 - no mismatch of specifications and program comments and the file created is usable for readable code for the FDA

Automated self-documenting Programs

- A specially marked comment / specification could look as follows (below)
- Using special keywords to mark comments
- Program collects all specially marked comments (also from called files / macros)

```
/*_START_SPECS
```

```
1. Read in data where xxx and yyy
```

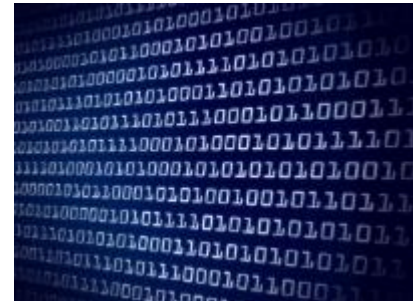
```
2. Derive zzz as follows
```

```
- xxxxxxxx
```

```
- yyyyyyyyyy
```

```
...
```

```
_END_SPECS*/
```



Automated self-documenting Programs

- The layout of the output (*.txt / *.pdf®) files looks as follows:

Programming Specification For XXXXXXXXXXXX.SAS®

1

Program Name : /path/xxxxxxx.sas®

Date/Time : DDMMYY/HH:MM

Specs xxxxxx

Specs are created
before the code!?

- The specifications between the _START_SPECS and _END_SPECS tags are printed one to one, i.e. formatting / layout etc. can be done within the comments
- In case the macro is called within a program which contains no marked comments, no *.txt or *.pdf® files are produced within the respective folder
- Some reporting programs include other dependent macros. e.g. test.sas® calls test_macro.sas® which is stored in another folder of e.g. the respective directory
- → output specification files can be combined / sent / linked etc etc

Automated Output / Table Shells

Problem

- Maintaining table shells in e.g. Statistical Analysis Plans (SAPs) on manual basis is inefficient
- The actual table / output produced, often looks different than the mock-up table that was defined

Solution

- Automated table shells created at run-time of a program can be used for inclusion in the SAP.
- This is done by replacing numbers e.g. with “x” at the time of the table / output creation
 - that table shells / mock-ups are always up-to-date
 - no manual copying to another document is needed
 - no mismatch of the actual table shell vs the mock-up table in the SAP.

Automated Output / Table Shells

- The program to automatically create mock-up tables is called at the very end of a reporting program, i.e. after the actual report object was produced
- The macro reads in the physical report object produced and creates a so called “dummy” output in the same directory where the report object is stored
- Dummy in this sense means that all digits (0-9) are replaced with “x” or any other character in the “body part” of the report object
- The suffix “_dummy” is automatically added to the original report object file name
- The macro features four parameters:
- **PAGEONEONLY / REPLACESTRING / COLUMN1 / BODYPART**
- In case no report object is produced, e.g. due to a program error, the macro recognizes that the output was not produced and skips the creating of the dummy-report object

Automated Output / Table Shells

1 Title Block
 ----- <- standard line
 2 Column Header(s)
 ----- <- standard line
3 Body
xxxx
 ----- <- standard line
 4 Footnote Block



1 Title Block
 ----- <- standard line
 2 Spanning Header
 ----- <- spanning header line
 3 Column Header(s)
 ----- <- standard line
4 Body
xxxx
 ----- <- standard line
 5 Footnote Block

Summary of xxxxx
 Protocol(s): AAAAAA
 Analysis Population: SAP (N=AAA)

 Subgroup: Female

Drug A	Drug B	Total
N=..	N=..	N=..
No. (%)	No. (%)	No. (%)

 Observation Time (Months)

Mean	xx.x	xx.x	xx.x
SD	xx.xx	xx.xx	xx.xx
Median	xx.x	xx.x	xx.x
Min	x	x	x
Max	xx	xx	xx
n	xx	xx	xx

Number Of Patients Observed

> x Months	xx (xx%)	xx (xx%)	xxx (xx%)
> x Months	xx (xx%)	xx (xx%)	xxx (xx%)
> x Months	xx (xx%)	xx (xx%)	xxx (xx%)
> xx Months	xx (xx%)	xx (xx%)	xxx (xx%)

 Footnote aabb

PAGEONEONLY
REPLACESTRING
COLUMN1
BODYPART

Study / Project Landing Zones

Problem

- In each study / project, teams need to maintain links to lots of documents (often done individually)
- Often it is also quite difficult to perform hand-overs of studies and projects
- We also work on different platforms, we have many documents, and many functions in a study / project contribute to the bigger picture

Solution

- Shared sense of purpose.
- Have a “one-stop shop wiki” for a project / study, e.g. a HTML-page with all study / project-related information and links
 - everyone contributes to such a page
 - ongoing updates are made in a timely manner
 - the page functions as a “one-stop shop”
 - all available links and documents etc. are just one-click away for everyone working in the study / on the project

Study / Project Landing Zones

- Advantages:
 - Functions as the central platform for all Project/Study- related information
 - Contains links to all Project/Study- relevant information sources
 - Pre-defined structure, covering all aspects of a Project/Study
 - Created in order to support transparency and knowledge sharing
 - First area for all Team Members users to go first
 - Making usage of the general wikipedia principles possible for Projects / Studies
 - Automated emails ideal (watching pages)
 - Linking of ad hoc request – tickets ideal (& categorization)
 - Building a Project/Study knowledge-base
 - Supports cumulative information gathering
 - Central maintaining of links and information (not on individual basis)
 - Reduces the time of users searching for information and knowledge
 - Collaborative platform where individual contributors build the «big picture»
 - Big help for onboarding / transition of studies and projects

Importance of Standards

...Analysis concepts vs TLG layouts...

Problem

- Rare diseases
- Special wishes of stakeholders
- Code sharing
- Re-inventing the wheel for table shells and outputs
- ...
- → These problems arise when there are no standards in place

Solution

- Have an established standards group
- Statistical programmers should perform active code sharing
- Code collaboration platform should be in place
- “education” of stakeholders to adhere to standards
 - faster filings
 - faster delivery of clinical study report outputs
 - Less programming effort

Importance of Standards



- Standardize as much as possible
- „education“ of stakeholders to adhering to these
- Company – specific reporting tool can focus on the standards
- Helps contributing to faster filings
- Helps stakeholders to get familiar with concepts and layouts
- Problem: when is something becoming „standard“?
- Standards in place → reporting tool can support the implementation of these
- Avoid „re-inventing the wheel“

Summary & Outlook & Further Thoughts

- Small and simple things can help with automation and with efficiency
- Time-savings will become more important in the future
- Data will grow constantly
- „focus on the right things“
- Automise and optimize standard processes in order to be able to have time focussing on other things
- Actively perform scope control
- Talk to your stakeholders



Summary & Outlook & Further Thoughts

- Interactive displays vs production of outputs
- Removal of manual steps → reduction of possible errors
- Push analyses to stakeholders by providing them with a „framework“
- Right tool for the right analysis
- Statistical Programmers upskilling



Q & A



Doing now what patients need next