

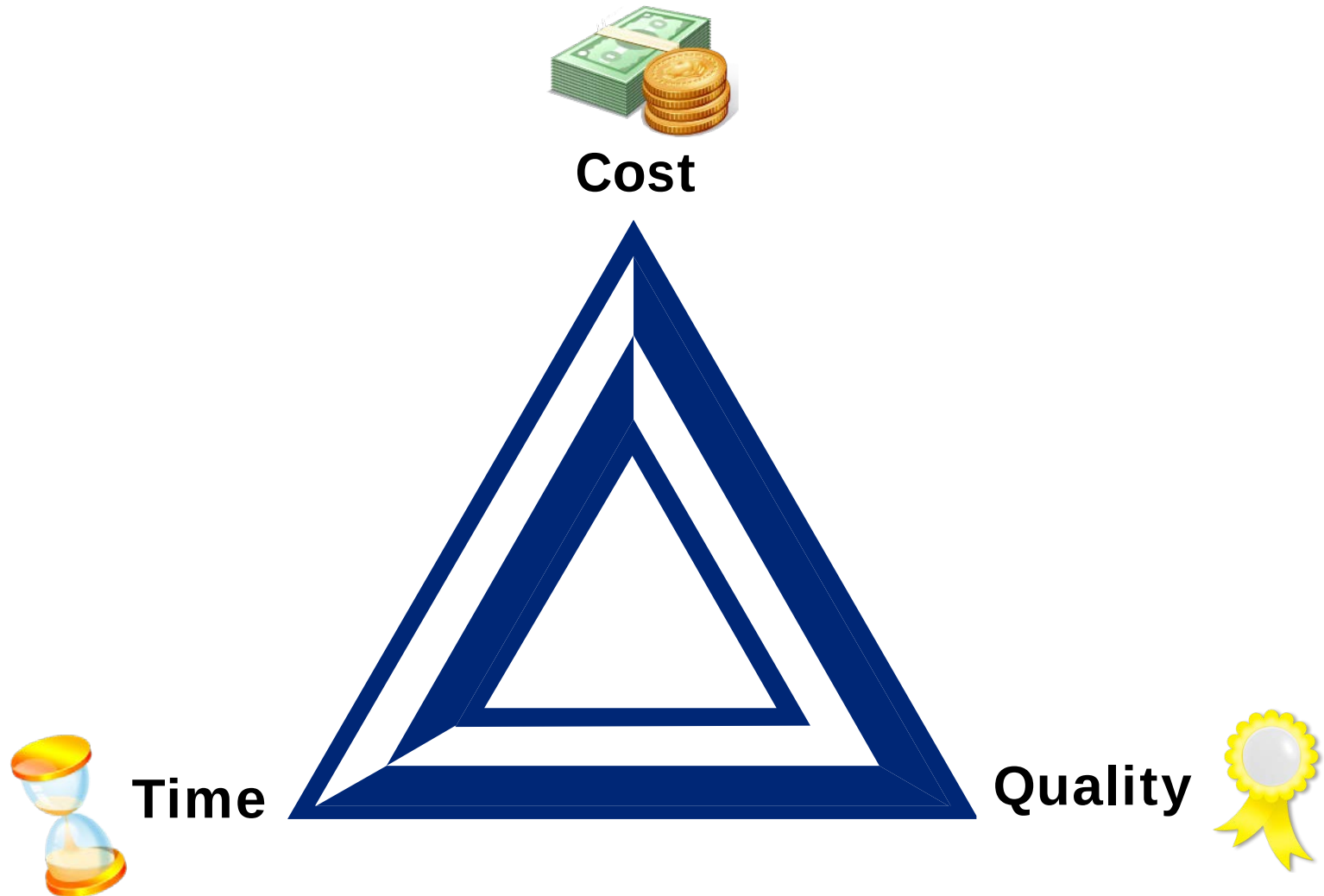
Standard Reporting Engine - StaR -

Pekka Heikkilä, CEO
StatFinn Oy

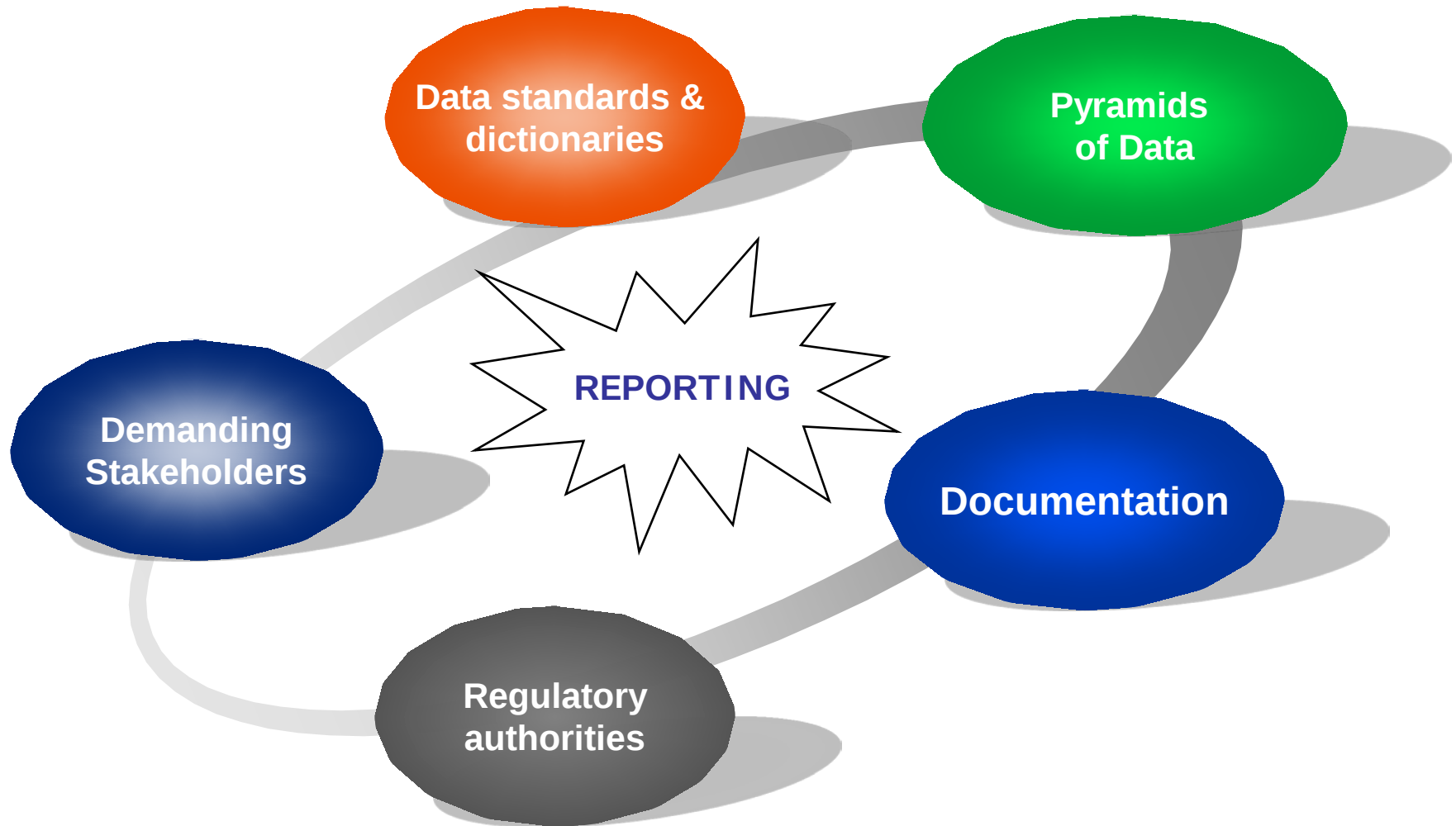


- Dilemma – WHERE IS THE PROBLEM
- Suggestion – WHAT IS THE DESIGN
- StaR is born - HOW DOES IT WORK
- Challenges – CHANGE MANAGEMENT

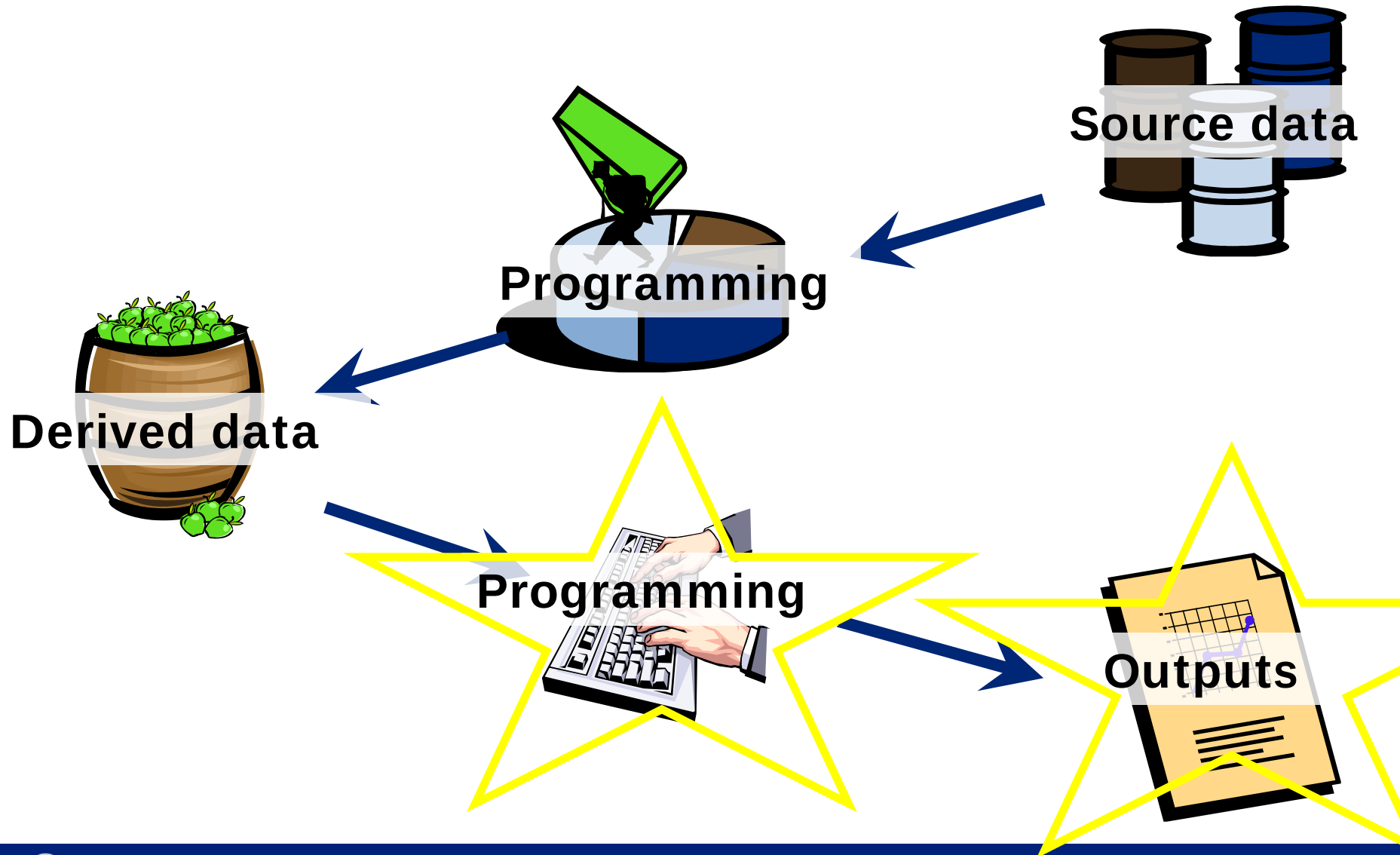
Project triangle



Additional challenges



Process flow



StaR

We all wrote our first macro which was followed by another one and one day we had our macro library.

This turned to a vision of standard reporting system, “red button” as I call it. What happened to this vision of yours?

StaR is mine, it is not complete, but it is still alive. There have been and still are challenges to conquer.

This is my story of obstacles and how they have been tackled.

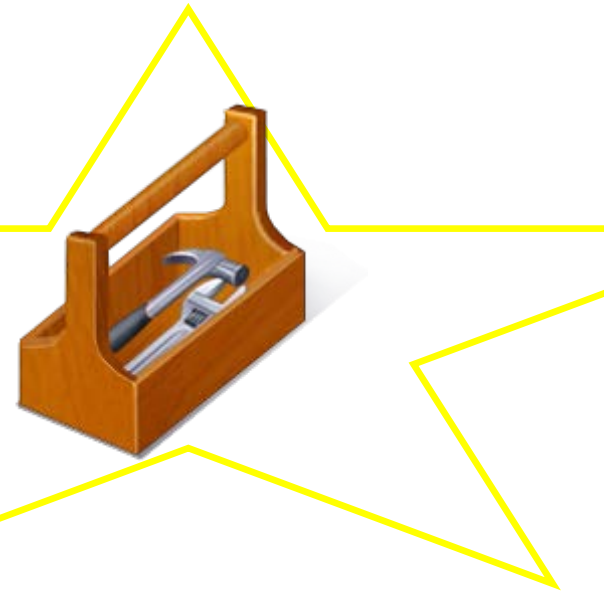
StaR

Not a black box



but

a tool box



StaR

96

Module I: StaR engine

- Study/Project coordination
- Environmental settings
- Titles and footnotes
- RTF layout specifications



0/1/4

Module II StaR outputs:

- Summary statistics for categorical data
- Descriptive statistics for continuous data
- Database listings
- Graph templates and tools



Module III StaR report:

- Combined appendix with common header and footer
- Table of Contents
- Study/project validation report



Module I macros

- Environmental macros
 - Are used for global settings
 - Paths/Libraries (STARTINIT)
 - Titles/Footnotes (READ_TLF, TTL_FTN)
 - Layout of output (RTFSTYLE, TRT_LABELS, PAGING)
- Supportive macros
 - Handle routine tasks
 - Number of observations (NOBS)
 - Variable type and existence (VAREXIST, VARNUMERIC)
 - Format creation (FMTEXIST, CREATE_FMT)
 - Combination of variables (VARCOMB)

Module II macros

- Summary statistics for categorical variables (CATEBY)
 - Counting frequencies and percentages
 - Printing directions of by variables
- Summary statistics of continuous variables (CONTIBY)
 - Calculating descriptive statistics (N, Mean, Std, Min, Median, Max)
 - Additional statistics (Quartiles, Geometric mean, CV%)
 - Controlling for decimals and alignment

Module II macros

- Graphical options and settings (CATEBY)
 - ⌚ Graphical options (GOPTIONS)
 - ⌚ Symbols and legend (SYMBOL, LEGEND)
 - ⌚ Axis statements with automated minimum and maximum (AXIS)
 - ⌚ Generate output with single or multiple graphs per page (GREPLAY)
- Data listings (DATALIST)
 - ⌚ Data listings (Chp. 16.2 from ICH E3)
 - ⌚ Individual subject data listings (Chp. 16.4 ICH E3)
 - ⌚ Listing all data from a given folder
 - ⌚ Controlling for a single data and variable

Module III Macros

- Generation of appendices (COMBINE_RTF)
 - Combining RTF outputs
 - Generating TOC
 - Common header and footer
 - Page numbering
- Coordination and QC/QA file
 - Program name and programmer
 - Validation level and status
 - Titles and footnotes

Program outline

```
/* Common header: Function, Input, Output  
(Programmer, Date, Modification) */
```

%startinit

```
%startinit(version=1,0);
```

%read_tlf

```
%read_tlf(version=1.0);
```

%ttl_ftn

```
%ttl_ftn(version=1.0, midle=Test study);
```

%rtfstyle

```
%rtf_style(version=1.0);
```

```
ODS RTF FILE="&rtffile." STYLE=StatFinn_rtf BODYTITLE;
```

%contiby

```
%contiby(version=1.0,  
          indata=temp1, var=sex age weight height bmi,  
          downvar=_var_, acrossvar=treat);
```

```
ODS RTF CLOSE;
```

Documentation

- Macro coordination sheet
 - Name and function
 - Modification history
- Macro specification document
 - Assumptions and usage notes
 - Description of parameters and their default values
- Macro self documentation
 - Call a macro without parameters will display each parameter with default values

Version control

- Environmental version control based on open source SVN (Subversions) software
 - Maintain current and historical versions of files/folders
 - Commits as true atomic operations
 - Support branching, tagging and locking
 - Path based authorization
- StaR macros under restricted/version controlled repository
- Selected version of macros assigned as “externals” for each project

Version control

- The SAS macros are designed with version controlling
- The version control allows to update the system without compromising the backwards compatibility
- Each major macro has a *coordination* macro which then calls the correct version of the main macro
- The coordination macro guides the user to use the most current version number and gives instructions of available parameters for the macro
- When a valid version number is given the coordination macro calls the correct macro with the given parameters

Example of version control

Original coordination macro: rtfstyle.sas

```
/******  
TASK:      Creates an TEMPORYRY RTF TEMPLATE  
AUTHOR:    Pekka Heikkilä  
DATE:      15AUG2008  
*****/  
  
%MACRO RTFSTYLE(version=);  
  
%IF &version.= 1.0 %THEN %DO;  
    %RTFSTYLE_V1(version=&version.);  
%END;  
  
%ELSE %DO;  
    %PUT Version number is needed to run &sysmacroname.;  
    %PUT Current version for &sysmacroname is 1.0.;  
%END;  
  
%MEND rtfstyle;
```

Example of version control

Original main macro: rtfstyle_v1.sas

```
/******  
TASK:                Creates an TEMPORARY RTF TEMPLATE  
AUTHOR:              Pekka Heikkilä  
DATE:                15AUG2008  
PARAMETERS:  
    version          - Version number (current = 1.0)  
OUTPUT:              STATFINN_RTF template  
*****/  
%MACRO RTFSTYLE_V1(version=);  
%PUT Running &sysmacroname. macro version = &version.    ;  
  
PROC TEMPLATE;  
    DEFINE STYLE StatFinn_rtf;  
    PARENT =styles.rtf;  
    REPLACE Body FROM Document /  
        topmargin    = 2 cm bottommargin = 1.8 cm leftmargin  = 4.5 cm rightmargin =  
        3.7 cm;  
END; QUIT;  
  
%MEND rtfstyle_v1;
```

Example of version control

Modification to coordination macro: rtfstyle.sas

```
/******  
TASK:      Creates an TEMPORYRY RTF TEMPLATE  
AUTHOR:    Pekka Heikkilä  
DATE:      15AUG2008  
MODIFICATION: - version 1.1 added by Pekka Heikkilä  
*****/  
  
%MACRO RTFSTYLE(version=);  
  
%IF &version.= 1.0 OR &version.=1.1 %THEN %DO;  
    %RTFSTYLE_V1(version=&version.);  
%END;  
  
%ELSE %DO;  
    %PUT Version number is needed to run &sysmacroname.;  
    %PUT Current version for &sysmacroname is 1.1.;  
%END;  
  
%MEND rtfstyle;
```

Example of version control

Modification to main macro: rtfstyle_v1.sas

```
...
MODIFICATIONS:      - Version 1.1 Pekka Heikkilä:
*****/

%MACRO RTFSTYLE_V1(version=);
%PUT Running &sysmacroname. macro version = &version.    ;

PROC TEMPLATE;
  DEFINE STYLE Wares_rtf;
  PARENT =styles.rtf;
  REPLACE Body FROM Document /
% IF &version = 1.0 %THEN %DO;
    topmargin    = 2 cm
%END;
%ELSE %DO;
    topmargin    = 1.5 cm
%END;
    bottommargin = 1.8 cm leftmargin  = 4.5 cm rightmargin = 3.7 cm;

END; QUIT;
%MEND rtfstyle_v1;
```

Example of version control

Modification to coordination macro: rtfstyle.sas

- version 1.1 added by Pekka Heikkilä

- version 2.0 added by Pekka Heikkilä

*****/

```
%MACRO RTFSTYLE(version=,orientation=LANDSCAPE);
```

```
%IF &version.= 1.0 OR &version.=1.1 %THEN %DO;
```

```
    %RTFSTYLE_V1(version=&version.);
```

```
%END;
```

```
% IF &version.= 2.0 %THEN %DO;
```

```
    %RTFSTYLE_V2(version=&version.,orientation=&orientation.);
```

```
%END;
```

```
%ELSE %DO;
```

```
    %PUT Version number is needed to run &sysmacroname.;
```

```
    %PUT Current version for &sysmacroname is 2.0.;
```

```
    %PUT Parameters.;
```

```
    %PUT          VERSION    = Version number (current = 2.0);
```

```
    %PUT ORIENTATION = Orientation of the output ((PORTRAIT),  
    LANDSCAPE);
```

```
%END;
```

```
%MEND rtfstyle;
```

Advantages

Standardising routine and repeated tasks to give you more time to evaluate the results

Flexibility of the system allows you to have your own outputs merged into the system

Standardized outputs and study coordination guarantees quality

Version control guarantees backward compatibility

Modular system design allows updates and upgrades

Open source code for your own on demand development of the system

Challenges

Kiitos !

