GPP:Gambling with Performance and Productivity

PhUSE SDE, Frankfurt, 15 April 2015

Shafi Chowdhury

GPP Board, Shafi Consultancy Ltd.

Contents

- What are the challenges?
- How do these affect performance?
- How do these affect productivity?
- What actions can we take?
- Examples
- Summary

What are the challenges?

- Reduced timelines to complete tasks
- Increase in re-use of program code,
- Reduction in local programming teams
- Increase in centralisation of program development
- Experienced programmers required at short notice

How do these affect performance?

- Old code is re-used or code is shared so the task can be completed quickly
- If it takes long, is that because of:
  - Your skills
  - Your ability to work under pressure
  - Your ability to understand programs from others, when this is now common practice
  - The program you took over is badly written

How do these affect productivity?

- Time estimates for individual programs dictate what can be achieved in a given time period
- If individual programs take longer then:
  - Increase pressure on the team
  - Need to increase resource at last minute
  - Working late into the night to complete task
  - Delay has impact on client, management and other projects and morale

What actions can we take?

- Train individuals to improve their SAS skills
- Provide clear and easy to follow guidelines for programming, e.g. PhUSE GPP guideline http://www.phusewiki.org/wiki/index.php?title=Good_Programming_Practice
- Monitor compliance to ensure the whole programming team adheres to agreed style
- Improve quality of our programs so that they are easy to read, understand and maintain over time

What actions can we take?

- Define clear specifications
- Programs should follow GPP:
  - Header: who wrote it, why, anything special
  - Structured program: input, process, output
  - Comments: explain what the next part of the program does and reason if appropriate
  - Datasets: use meaningful names, do not overwrite, use different names
  - Indent: use consistent spacing

Examples

```
1
2  *Macro to create an ATTRIB statement from a CSV file *;
3
4  %macro x_attrib(path=, csv_file=);
5
6  %if %length(&path.)=0 %then %let path=%sysfunc(pathname(pgm));;
7
8  data x_attrib;
9  length varname $8. varlbl $60. datatype $4. varlen 8. varfmt $10.;
10 infile "&path.\&csv_file." dsd dlm=',' lrecl=1000 pad firstobs=3 missover end=eof;
11 input varname $ varlbl $ datatype $ varlen varfmt $;
12
13 vnam=10;
14 vfmt=20;
15 vlbl=40;
16
17 if _n_=1 then put '  attrib' @@;
18 if eof and varlbl = ' ' then put @vnam ';';
19
20 if varlbl ne '';
21
22 varname = lowcase(varname);
23 if varfmt ne '' and index(varfmt,'.')=0 then varfmt=trim(varfmt)||'.';
24
25 lbllen=length(varlbl);
26 if lbllen gt 40 then put @3 '*' lbllen @@;
27
28 put @vnam varname @vfmt 'format = ' varfmt @vlbl 'label = "' varlbl '"';
29
30 if eof then put @vnam ';';
31 run;
32
33 %mend x_attrib;
34
```

Examples

```
/*=====================================================================
*
* STUDY                : xxxx.xx                CREATION DATE: 13AUG2007
*
* PROGRAM              : x_attrib.sas
*
* PROGRAMMER           : Shafi Chowdhury
*
* DESCRIPTION          : Create an ATTRIB statement based on the variable
*                          descriptions specified in the ADS specification.
*
* COMMENTS             : Take care that in the INFILE statement FIRSTOBS
*                          points to row in the CSV file containing the
*                          first variable.
*=====================================================================*/

OPTIONS MPRINT NOMLOGIC NOSYMBOLGEN SOURCE SOURCE2 NOTES NOFMTERR MSGLEVEL=i;

  ********************************************************************;
  *** Create an ATTRIB statement from a CSV file created from     ***;
  ***   the ADS description in the TSAP copied to Excel.         ***;
  ***                                                            ***;
  *** NOTE: eof is referenced twice incase the last record is    ***;
  ***         deleted due to missing variable label.             ***;
  ***                                                            ***;
  *** a. Define positions where the VARIABLE name, FORMAT and    ***;
  ***      LABEL should be positioned.                           ***;
  *** b. On the first observation write ATTRIB in the LOG.       ***;
  ***    If its the last observation then write the semi-colon   ***;
  ***      to mark the end of the ATTRIB statement.              ***;
  *** c. Only keep observations if the variable has a label.     ***;
  *** d. If formats do not have a '.', then add a '.' at the end ***;
  *** e. Flag variables with labels longer than 40 characters as ***;
  ***      they will have to be shortened.                       ***;
  ***    Write the VARIABLE  FORMAT = ...  LABEL = ".." statement***;
  ***                                                            ***;
  ********************************************************************;
```

```
%MACRO x_attrib(PATH=, CSV_FILE=);

  %IF %LENGTH(&path.)=0 %THEN %LET path=%SYSFUNC(PATHNAME(pgm));;

  DATA x_attrib;
    LENGTH varname $8. varlbl $60. datatype $4. varlen 8. varfmt $10.;
    INFILE "&path.\&csv_file." DSD DLM=',' LRECL=1000 PAD FIRSTOBS=3 MISSOVER END=eof;
    INPUT varname $ varlbl $ datatype $ varlen varfmt $;

%*a*;
    vnam=10;
    vfmt=20;
    vlbl=40;

%*b*;
    IF _N_=1 THEN PUT '  ATTRIB' @@;
    IF eof AND varlbl = ' ' THEN PUT @vnam ';';

%*c*;
    IF varlbl NE '';

%*d*;
    varname = LOWCASE(varname);
    IF varfmt NE '' AND INDEX(varfmt,'.')=0 THEN varfmt=TRIM(varfmt)||'.';

%*e*;
    lbllen=LENGTH(varlbl);
    IF lbllen GT 40 THEN PUT @3 '*' lbllen @@;

    PUT @vnam varname @vfmt 'FORMAT = ' varfmt @vlbl 'LABEL = "' varlbl '"';

    IF eof THEN PUT @vnam ';';
  RUN;

%MEND x_attrib;
```

Examples

```
76
77  /* Example:
78
79  %x_attrib(csv_file=basco_variables.csv);
80
81  Example CSV file:
82
83  Variable,Label,Datatype,Length,Format,Example,Comments
84
85  PTNO,Patient number,Num,8,10,301,Patient number as it is recorded on the CRF
86  CENTRE,Centre ,Num,8,10,4902,Centre number as recorded in the database
87  CENTREP,Pooled centre number,,,8,1,If necessary
88
89
90  Example attrib statement in the LOG:
91
92    ATTRIB ptno      FORMAT = 10.        LABEL = "Patient number "
93           centre    FORMAT = 10.        LABEL = "Centre "
94           centrep   FORMAT = 8.         LABEL = "Pooled centre number "
95           ;
96
97  */
98
99    *******************************************************************;
100   *** Clean up WORK area                                          ***;
101   *******************************************************************;
102
103 PROC DATASETS LIB=work MEMTYPE=data NOLIST;
104   DELETE x_attrib;
105 QUIT;
106
107   *******************************************************************;
108   ***                                                             ***;
109   *** End of program :                                            ***;
110   ***                                                             ***;
111   *******************************************************************;
```

Summary

- Do not gamble with performance and productivity
- Following GPP reduces risk in the future
- Define standards
- Ensure everyone follows the standards
- PhUSE Wiki for guideline and more details:

  http://www.phusewiki.org/wiki/index.php?title=Good_Programming_Practice

Questions?

Shafi Chowdhury

GPP Board, Shafi Consultancy Ltd.

shafi@shaficonsultancy.com