PharmaSUG 2015 - Paper TT10

Essential Guide to Good Programming Practice

Shafi Chowdhury, Shafi Consultancy Limited, London, United Kingdom

Mark Foxwell, PRA Health Sciences, Reading, United Kingdom

Cindy Song, Sanofi-Aventis, Bridgewater, New Jersey

ABSTRACT

Due to increased needs of the pharmaceutical industry and the success of SAS®, an ever growing number of programmers are joining the industry all over the world and inevitably moving from one organization to another. Not to forget about code that is shared across companies and with the regulatory agencies. Therefore, to ensure continuity and allow code to be maintained over time, it is essential for organizations that a programming standard is followed.

If there is an industry standard GPP guideline, that is widely recognized and adopted; then organizations can be sure that when new programmers join, they can start to both use existing code with ease, and develop new code that others already in the organization can easily follow. In addition of course their existing programmers can share code easily with each other.

PhUSE GPP steering board has developed a GPP guideline that has been out for more than a year. It highlights the main standards that should be followed and what should be avoided. The steering board consists of very experienced programmers from large and small CROs and Pharmaceutical organizations, so they have seen what works and what causes problems. These few easy to follow standards offer the industry a simple way to ensure we continue to develop code that can be easily maintained and shared over time, saving time and cost. This new guideline could either be understood as fundamental principles that can be extended by company specific rules or can just be used as is.

INTRODUCTION

The need for a Good Programming Practice (GPP) Guideline is more important now than ever before. At a time of reduced timelines, limited resources, increase in remote teams, the practice of sharing code is more prevalent and necessary than ever before. However, in order to make the sharing of code effective, it is important that the code is easy to read, easy to understand and easy to update. If any of these three simple tests fail, then sharing of code is less beneficial than rewriting it from scratch, it is simply being done for the sake of sharing, with little or no actual benefit.

This paper describes the key steps a programmer needs to undertake, and organisations need to set up, in order to ensure the programs produced not only do what they are programmed to do, but that they continue to be useful in the future. Re-using programming code is one of the most efficient ways to resolve the problems with resource and short timelines. It can also help to drive standards and ensure consistent outputs are produced across studies.

Defining a clear and easy to follow guideline is essential for any organisation wishing to improve the programming standards within their organisation. However, successful implementation of the guideline and monitoring its compliance is often where all the hard work becomes undone. This paper therefore summarises the feedback from PhUSE conference GPP discussion club to provide guidance on some of the strategies used to successfully implement a good programming practice guideline.

WHAT IS GPP?

Good Programming Practice Guideline is a set of rules developed over time that have shown to produce robust programs that not only produce correct results, but are easy to read, understand and maintain over time. The guideline defines not only how to develop the program, but also how to test and maintain them over time so they continue to produce correct results.

The basic principles behind GPP Guidelines are:

- The rules are easy to follow
- Improves clarity and readability of programs
- Improves quality and reliability of programs
- Makes programs easy to update

Although most Pharmaceutical organisations have their own GPP Guideline that can range from three to fifty pages, these aims are the cornerstone behind them all.

INDUSTRY STANDARD

Accepting that the main aims behind the many different GPP Guidelines are the same, it then follows that an industry standard guideline is possible. An industry guideline is long overdue, but as the globalisation and remote working practices are part of the normal working practices these days, this guideline is more in need now than ever before. An industry guideline will allow:

- Programmers to follow a consistent programming style, not change with each employer
- New team members to pick up each other's code at short notice and understand the programs
- Employers to expect new recruits to program in the same style as the existing members of the team
- CROs to develop code that can be used for multiple clients and ensure the client standards are followed
- Companies using multiple external partners to receive all the programs in a consistent style
- Independent review and validation to be performed more easily
- Regulators who review code to expect a consistent style, so it is easier for them to check
- A benchmark to be defined, against which all programs can be compared
- Generation of utility tools to check for compliance against a recognised industry standard

We believe that an industry standard can only be of benefit to the industry.

PROCESS DEFINITION

As with any standards, to be successful, it must be backed up by a defined company process. If this process is not implemented with the same rules and regulations as others, and is treated lightly, then the standards defined in the guideline will not be followed. They will simply become no more than electronic versions of shelf fillers gathering dust that some of the current guidelines suffer from. Implementation, compliance and checking compliance must therefore be part of a strict process.

Programs in the Pharmaceutical Industry often have two program life cycles. There is the study life cycle of defining requirements, developing, testing, and final quality check on final data. This is then followed by an extended life cycle where the program has to be updated for a publication or need to be re-run on new data for the same study. This can be thought of as the standard study life cycle.

There is a second company lifecycle that a process must also consider. This is where the final program from one study is then copied to another study. How this program is managed and validated must be clearly defined within the process. Without clear process definition, confusion can lead to a program that requires update for different studies being missed, and an incorrect result is reported. This is particularly important when there is a separate validation program, and that is also being copied and used by a second programmer in order to save time.

Receive Specification => Develop Program => Validate against specification => QC final data output

↕↕

Update Specification

Receive Specification => Copy an existing program => Validate against specification => QC final data output

GOOD PRACTICES

Once the process has been defined to support the implementation of GPP guideline and to monitor its compliance, it is important to define the good practices that the programmers should follow. It is important during this process that we always go back to the guiding principles.

IMPROVE CLARITY AND READABILITY

Clarity and readability is improved by appropriately documenting the program and having a structure and layout that makes it easy to look at. To achieve these, the program should have the following:

- Program header
    - Name and location of program
    - Date when program is first started / completed
    - Name of programmer
    - Purpose of the program
- Follow INPUT, PROCESS, OUTPUT structure, split up into distinct sections
    - Read in all external datasets at the top, use KEEP statements to clarify what comes from where
    - Process all data, derive variables, perform analysis and calculations as required
    - Produce final external datasets (use KEEP statement), or produce tables/ listings and figures
- Comments above data steps to say what is happening in the following steps
- Separate data steps and procedures by blank lines
- Unique and informative dataset names
- Only one SAS statement on each line
- Indentation of code within data steps and procedures using a consistent but defined number of spaces
- Using uppercase for SAS keywords, and lowercase for variable and dataset names

IMPROVE QUALITY AND RELIABILITY

- Defensive programming
    - Check for implausible values
    - Have a defined process for identifying, reporting and confirming resolution of data issues
    - Program for likely changes of data
    - Always program to handle missing values
- Informative comments and relate how derivations are programmed to the specification
- Independent programs for testing the output where appropriate
- Platform independent code
- Keyword parameters for macros so users know what is going into the macro call
- Options that highlight programming deficiencies, such as MSGLEVEL=I, which list common variables that are overwritten during a MERGE statement

EASY TO MAINTAIN / UPDATE

- Naming convention for programs
- Comments to state if something is study specific or if care should be taken during updates
- Clear structure that makes it obvious where to make an update without unintended impact elsewhere in the program
- Have a summary of the steps within the program at the top so that someone can see what is happening within the program, especially if it is a long program
- Use standard macros for common tasks
- Generate automatic programs based on meta data to ensure consistency across programs and studies

THINGS TO AVOID

Just like there are steps that we can take to improve our programs, there are also steps that we should avoid. Although often it is simply the opposite of the best practices, there are some practices that get a special mention. An easy check is always to go back to our first principles again, clarity, readability, quality, reliability and easy to maintain. Anything that adversely impacts any of these should be avoided.

- Excessive documentation so the program loses the flow and becomes unreadable
- Calling procedures without defining input dataset names
- Writing macros without keyword parameters
- Overwriting temporary datasets
- Keeping variables that are not needed for analysis
- Lack of white space
- Reference to external files dotted throughout the code
- Use of many data steps when a lot of updates can be done on one read of the data

CONCLUSION

Guidelines for Good Programming Practices are often long with many theories and justifications for each statement. However, this often detracts from the key points, and can lose the audience it is trying to engage. Without any monitoring of compliance, the guidelines with all its good intentions unfortunately do not lead to the success of great reusable programs we all expect.

Following the 80/20 approach, a list of key do's and don'ts a single page and monitoring compliance with zero tolerance is perhaps the most pragmatic approach to take. This may sound harsh, but not complying can lead to programs that are difficult to maintain over time, costing time and causing delay.

The key things to follow therefore are:

Do's	Don't
Use defined program naming conventions	Add excessive documentation within the program
Use a program header	Overwrite dataset names
Structure program – Input, Process and Output	Create long datasets
Use informative comments at appropriate places	Avoid excessive use of SQL
Use unique but informative dataset names	Turn off full disclosure of Errors, Warnings and Notes to the LOG
Use one statement per line	Manually edit the LOG or OUTPUT
Indent consistently within the program	
Use defensive programming code	
Use SAS options to reduce possible overwriting of variables, e.g. MSGLEVEL=I	
Check LOG, including for Cartesian Product when using SQL	

ACKNOWLEDGMENTS

We would like to thank the PhUSE GPP working group for all their support in developing the guideline and sharing this information.

RECOMMENDED READING

- PhUSE Good Programming Practice Guideline:
  http://www.phusewiki.org/wiki/index.php?title=Good_Programming_Practice

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name:	Shafi Chowdhury
Enterprise:	Shafi Consultancy Limited
Address:	Regus House, Highbridge Industrial Estate, Oxford Road
City, State ZIP:	Uxbridge, UB1 8HR
Work Phone:	+44 1895 876 533
E-mail:	shafi@shaficonsultancy.com
Web:	www.shaficonsultancy.com